UNCLASSIFIED

SECUR

AD-A259 485

RT DOCUMENTATION PAGE

1a. RE
UN

1b. RESTRICTIVE MARKINGS
None

2a. SE
N/A

3. DISTRIBUTION / AVAILABILITY OF REPORT
Unlimited

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE
N/A

ELECTE
DEC 31 1992
S D
A

4. PERFORMING ORGANIZATION REPORT NUMBER(S)
FR.0131-00

5. MONITORING ORGANIZATION REPORT NUMBER(S)
This document has been approved for public release and sale; its distribution is unlimited.

6a. NAME OF PERFORMING ORGANIZATION	6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION
JRS Research Laboratories Inc.	N/A	U.S. Army, Laboratory Command ET&D Laboratory

6c. ADDRESS (City, State, and ZIP Code)	7b. ADDRESS (City, State, and ZIP Code)
Orange, CA 92665-4121	Fort Monmouth, NJ 07703-5000

8a. NAME OF FUNDING / SPONSORING ORGANIZATION	8b. OFFICE SYMBOL (If applicable)	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER
U.S. Army LABCOM	SLCET	Contract # DAAL01-89-C-0912

8c. ADDRESS (City, State, and ZIP Code)	10. SOURCE OF FUNDING NUMBERS			
	PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT ACCESSION NO.
Fort Monmouth, NJ 07703-5000				

11. TITLE (Include Security Classification)
Final Report for the SBIR Phase II Program PC-Based VHDL System

12. PERSONAL AUTHOR(S)
Runner, Deborah W.

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
FINAL	FROM 890831 TO 920330	920330	

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	VHDL, PC-based VHDL, CAD, CAE, simulation, design automation
06	33	04	

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

The requirement to use VHDL as a language for describing hardware in DoD systems forces government contractors to acquire or build software systems to support VHDL processing. The existing software systems run on expensive computer equipment. This project provides a low-cost hardware design workstation based on a PC and it encompasses the requirements of a significant segment of the design community. In Phase II, JRS ported the Intermetrics IEEE 1076 VHDL System to a 386 PC Unix environment. JRS integrated additional tools with the PC/VHDL system so the workstation provides a communications interface to VAX and SUN environments, provides RTL-level design tools, provides several additional simulators, and provides a menu-driven user-interface to the tools, complete with HELP capabilities. The PC tool is extremely fast, user friendly, and does not limit the size of VHDL designs. Extensive documentation and user scenarios are included.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT	21. ABSTRACT SECURITY CLASSIFICATION
☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT. ☐ DTIC USERS	UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code)	22c. OFFICE SYMBOL
Fred Glickman	908-544-4963	SLCET-IC

DD FORM 1473, 84 MAR

83 APR edition may be used until exhausted.
All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

JRS RESEARCH LABORATORIES INC.
1036 W. TAFT AVENUE, ORANGE, CALIFORNIA 92665-4121 / TELEPHONE (714) 974-2201

Document Number FR.0131-00

Scientific and Technical Report,
Final Report
for the
SBIR Phase II Program
PC-Based VHDL System

Contract No. DAAL01-89-C-0912
Project No. CT-9R4111-CTCT
CDRL SEQUENCE NO. A003

JRS Research Laboratories Inc.
1036 West Taft Avenue
Orange, CA 92665-4121
(714) 974-2201

30 March 1992

Final Report for 31 August 1989 through 30 March 1992

Prepared for:

U.S. Army, Laboratory Command
ET&D Laboratory
Fort Monmouth, NJ 07703-5000

Contents

List of Figures

Chapter 1

Scope

1.1 Identification

This document summarizes the work done under the SBIR Phase II contract for the PC-based VHDL system and documents the success achieved in developing a PC/VHDL workstation that was delivered by JRS in January 1992.

The contract number is DAAL01-89-C-0912, project No. CT-9R4111-CTCT, covering the period from 31 August through 30 March 1992. This scientific and technical report, subtitled final report, is CDRL sequence number A003.

Chapter 2 contains a description of the project.

Chapter 3 lists the deliverables.

Chapter 4 describes the accomplishments.

Chapter 5 contains the conclusions and recommendations.

Chapter 6 lists the referenced documents.

The definition of the acronyms used in the document is contained in Chapter 7.

Appendix A shows the test results from executing 704 Intermetrics VHDL programs that were the IEEE VHDL test suite.

Appendix B shows execution timing to assemble, link, and simulate on the PC/Unix RTL toolset an ACVC test suite containing about 200 Ada programs.

Appendix C shows those same ACVC tests executing on a Sun 4.

Appendix D contains the Intermetrics study on characterization of VHDL features with regard to performance. Test data is included at the end of report to show the measurements for 31 VHDL test cases and the statistics for each VHDL feature tested.

1.2 Purpose

The requirement to use VHDL as a language for describing hardware in DoD systems forces government contractors to acquire or to build software systems to support VHDL processing. The existing software systems run on expensive computer equipment. The purpose of this project is to provide a low-cost system that is based on a PC and encompasses the requirements of a significant segment of the design community.

The SBIR Phase I effort focused on the system design, specification, and planning activities involved.

Phase II of this SBIR project proved that the Phase I report describing a PC-based VHDL version could, in fact, be implemented.

1.3 Configuration

Figure 1.1 shows the main overview of the PC/VHDL workstation. The modules involved execute on a PC AT 80386 system.

- VHDL design entities can be entered directly on the PC or from a remote network processor. Each design entity can be processed by the PC/VHDL analyzer which checks the syntax and, if correct, converts VHDL to IVAN database form. This form is used for model generation and simulator building blocks. Both structural and behavioral simulators can be built from the VHDL description. VHDL commands were kept very close to the original Intermetrics Sun-3 commands.
- Machine descriptions can be translated from VHDL into RTL Database Description (RDD) format from which a JRS fast simulator can be built. This simulator is faster and allows the user to check out behaviors of the machine.
- Another advantage of the RDD format is that the PC/VHDL workstation includes an RDD input editor that prompts the user with questions that are easy to answer. User responses are automatically for-

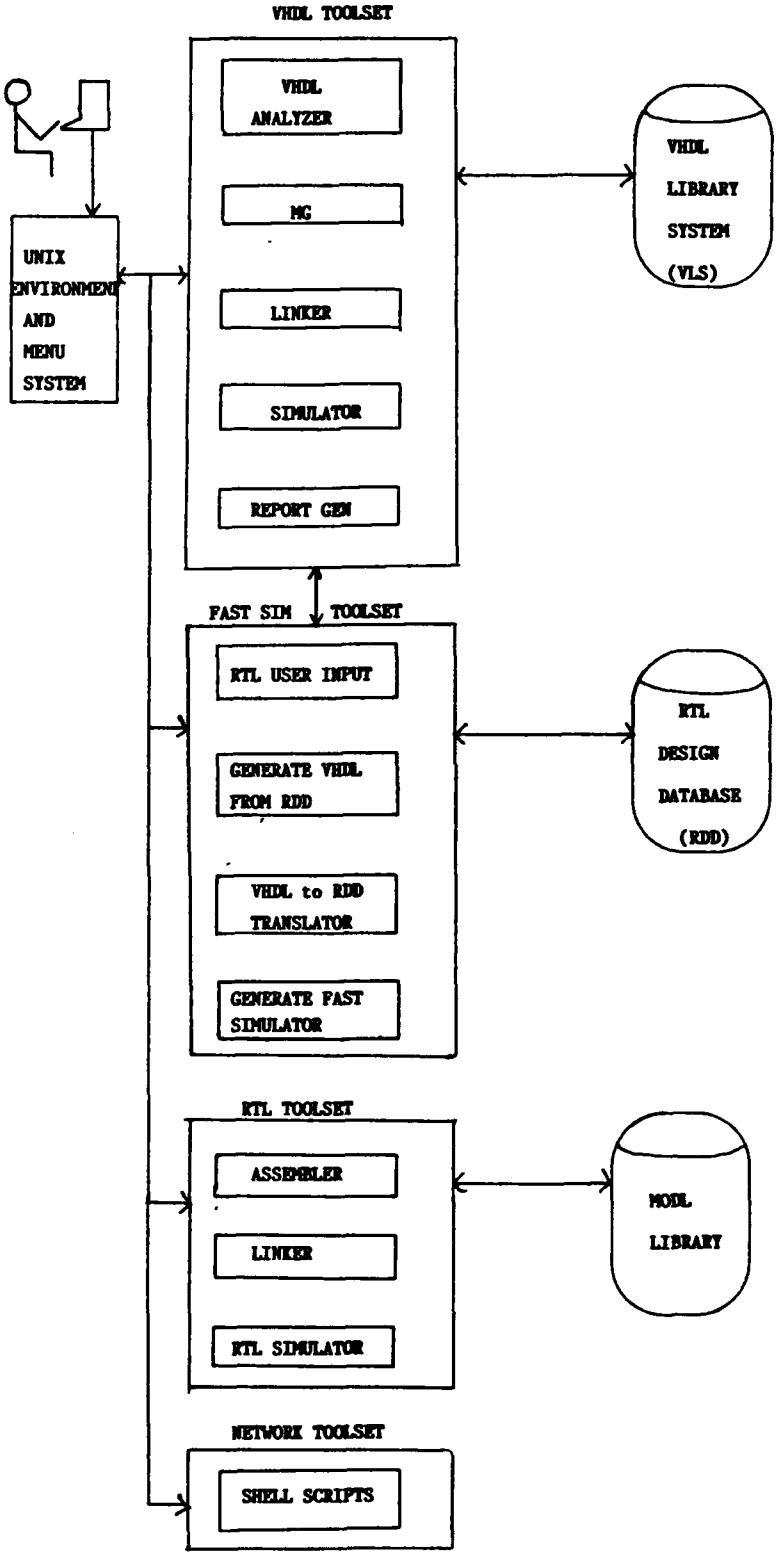

Figure 1.1: JRS PC/VHDL Workstation Overview

matted by the tool into an RDD description of the machine. The RDD form can then be translated into VHDL.

- The PC/VHDL system uses Unix which allows multi-processing to occur on the workstation. It also allows network resources such as disk space and printers to be accessed by the PC. This gives the user greater flexibility and even allows the PC/VHDL system to be driven by other processors on the network.
- An extensive RTL toolset is integrated with the PC/VHDL system to allow the user to design individual design entities and test them through an RTL language interface before they are embedded in a complete machine. The toolset includes assembler and linker capabilities and an RTL simulator provides interactive microcode debugging for users who need greater software support.

Configuration requirements for the PC/VHDL system are as follows:

- 386 PC with keyboard
- Memory expansion card
- 8 MB RAM
- CDC ST4702N SCSI 600 MB hard disk drive with CDC 3280C SCSI controller –or– 1 or 2 CDC 155 MB hard disk drives with ESDI controller
- TEAC 5.25" 1.2MB floppy disk drive
- VGA, EGA, or Hercules card and monitor
- 3COM Etherlink II 3C501

The PC/VHDL system runs in an Interactive Unix environment. The Interactive C compiler is necessary to support VHDL model generation. The Interactive Workstation Developer which includes the necessary environment can be purchased from the Interactive distributor, PGI Corporation (800-528-1415).

Chapter 2

Project Description

2.1 Background

The Intermetrics Sun-3 version 2.0 was used as the source code basis for the PC/VHDL. The version was stable, it was written in Ada, and it was compiled with the Verdix Ada compiler. JRS evaluated numerous run-time environments for Unix and Ada as described in *Evaluation of Runtime Environments* (refer to Section 6). JRS selected the PC Interactive Unix and Verdix PC Ada, and porting began.

2.2 Objectives

The objective of this contract was to implement the SBIR Phase I report which called for a lower-cost VHDL workstation. Some of the specific objectives of the project were:

- To provide a reduced-cost VHDL design environment.
  - Limited resource companies need reduced-cost VHDL to bid on VHDL contracts.
  - PC/VHDL would reduce hardware support costs to the government.
  - PC/VHDL would promote VHDL use in industry & academia.
  - PC/VHDL would enhance the use of VHDL in the large commercial design community.
- To provide a PC-based VHDL toolset to integrate with JRS IDAS.

- To port Intermetrics IEEE 1976 VHDL & support environment onto a low-cost platform.
  - Provide the same functionality on a PC as on a VAX.
  - Provide the best possible performance for a given configuration.
- To provide a network workstation environment for VHDL.
  - Allows VHDL support environment to share network resources.
  - Allows access to VHDL functionality via workstation-style user-friendly interface.
- To provide a dedicated VHDL server node to offload intensive VHDL processing.

These goals were met by providing three versions of the PC/VHDL system: a baseline version, delivered in March 1991; a network version, delivered in September 1991, and a network server, where JRS integrated an RTL toolset and a fast simulator toolset with the PC/VHDL tools and delivered them together as a menu-driven system in January 1992. Each PC version successfully executed the IEEE VHDL test suite of 700 programs. The later versions additionally executed nearly 200 software load modules that were developed on Sun and Vax IDAS systems and transferred over the network to run on PC/VHDL-generated simulators.

2.3 Features

The product consists of the following capabilities:

1. The complete Intermetrics IEEE 1076 VHDL Version 2.0 and all of its associated Sun 3 VHDL commands.
2. The complete RTL toolset including
   - A language for describing the RTL.
   - A language for describing the bus connections between individually programmable machines.
   - Tools to create an RTL assembler and linker.
   - Tools to create an RTL level simulator and debugger.
   - A tool to generate the bit settings for the microword used as input to the ROM.

3. The complete RTL-level Design Database (RDD) toolset including

   - Tools to input RDD machine descriptions.
   - Tools to convert RDD machine descriptions into VHDL descriptions.
   - Tools to convert VHDL behavioral descriptions into RDD descriptions.
   - Tools to generate VHDL behavioral simulators from RDD.
   - Tools to generate Ada fast simulators from RDD.

4. Network communications to perform the following tasks:

   - Save files from PC to Vax.
   - Get files from Vax to PC.
   - Save complete PC directory on Vax.
   - Restore the complete saved directory from Vax to PC.
   - Save files from Vax to PC.

5. User commands to perform the above 4 sets of work.

6. Several complete user's guides describing the above capabilities.

7. Numerous examples, including a sample RUI input session for the MACH5 machine.

8. Test programs, test data, and test results from the following:

   - Executing IEEE VHDL tests on the PC/VHDL system.
   - Generating a VHDL behavioral simulator for 3 different machines described in VHDL: Analytyx On-Board Processor (OBP); MACH5, a synthesized processor based on silicon compiler primitives; and the TI Aladdin processor.
   - Generating a VHDL behavioral simulator for the 3 machines described in RDD.
   - Generating an Ada fast simulator for the 3 machines described in VHDL.
   - Generating an Ada fast simulator for the 3 machines described in RDD.
   - Executing JRS-compiled microcode for the 3 machines on the VHDL behavioral simulator.
   - Executing JRS-compiled microcode for the 3 machines on the Ada fast simulator.

9. A manual that includes a description of how to interface microprograms and data to the simulators generated by the PC/VHDL RDD/RTL system.

10. Intermetrics results from characterizing VHDL features and generating useful analysis information for estimating disk size and CPU time based on a given test suite.

These capabilities fulfill the project plan outlined in the JRS *Plan for Remaining Development for the PC Based VHDL Design System* (refer to Section 6).

2.4 PC/VHDL Workstation Architecture

Figure 1.1 shows the various elements of the JRS Integrated Hardware Design Toolset. This section discusses the development stages and any problems that were faced during porting and integration. Instructions for installation, setup, operation, and backup of the tools can be found in the users' manuals listed in Section 6; in particular, the *JRS Integrated Hardware Design Toolset User's Manual*, the *Texas Instruments VHSIC RTL Toolset User's Guide*, and the *JRS RTL User Input User's Manual.*

2.4.1 Menu System

The JRS Integrated Hardware design menu system enables the user to access the various functional elements of the workstation from a top-level menu. Menu items allow the user to:

- Invoke PC/VHDL commands.
- Perform RTL input, conversion, and simulation.
- Execute database functions.
- Invoke network communications.
- Access an automated test system for several example machines.
- Request *help* information for each menu item.

Selecting a function from the main menu prompts for additional information or causes further menus to be displayed.

Figure 2.1: JRS Integrated Hardware Design Main Menu

```
-------------------------------------------------[jrs]
      JRS INTEGRATED HARDWARE DESIGN MAIN MENU
-------------------------------------------------

          [A] ADD/CHANGE USER:
          [C] CONVERT VHDL/RDD MENU
          [N] NETWORK COMMUNICATIONs MENU
          [R] RTL TOOLSET MENU
          [T] TEST SYSTEM MENU
          [U] RTL USER INPUT (RUI) PROGRAM
          [V] VHDL 1076 TOOLSET MENU

          [H] HELP                [X] EXIT/RETURN

     Enter Selection ==>

-------------------------------------------------
```

2.4.2 VHDL Toolset

The menu system for the JRS Integrated Hardware Design Toolset allows the user to select and execute the Intermetrics Standard VHDL 1076 support Environment tools. The elements of this PC/VHDL environment are the same as the Intermetrics VAX/VMS and the Sun VHDL systems with the exception of some low-level swap-handling routines that are written in the assembly language of the target machine. The Intermetrics version 3.0 Sun 3 software system was ported, module-by-module, feature-by-feature to the PC 386 environment.

The *Technical Memorandum, Evaluation of Runtime Environments* (see Section 6), documents the evaluation performed by JRS for an appropriate runtime and development environment to support PC/VHDL. As a result of this study, the Interactive Unix system was chosen as the operating system. This development environment includes a C compiler which is used by the VHDL simulation generation step. (Section 1.3 describes the configuration environment in detail.) JRS evaluated and acquired the Verdix Ada compiler for the PC. This compiler was used to recompile the Sun 3 VHDL software modules onto the PC. Some problems were encountered with the Verdix Ada compiler regarding tasking termination and temporary file closing, but work-arounds were found for all of the problems.

JRS was concerned with the following objectives regarding the execution of VHDL commands on the PC:

1. Obtaining the best possible performance for the given configuration.
2. Tailoring the software such that useful VHDL models of interesting functionality would be accepted as input and utilized with reasonable system performance.

These objectives have been met because the hardware/software configuration discussed in Section 1.3 provides a workstation that delivers execution speeds much faster than a DEC MicroVAX-II and execution speeds nearly as fast as the Sun 4 Sparcstation.

2.4.3 RTL Toolset

JRS ported the Texas Instruments VHSIC Register Transfer Language (RTL) Toolset from the VAX/VMS environment to the PC Unix environment. It was originally written by TI in VAX Pascal. JRS recompiled the source code using Meridian Pascal, changing VAX-peculiar code structures to "vanilla-Pascal" where necessary. The RTL tools have been included in the PC workstation because they provide a rich hardware/software design environment for building individual components of a machine and testing those components with user-written RTL programs. Additionally, complete machines can be represented in the RTL Design Database (RDD) form and compiler-generated microcode can execute on the RTL interactive debugger/simulator that is part of the toolset. JRS has provided translators between RDD and VHDL to give the user more flexibility.

The RTL Toolset menu permits the user to execute the Texas Instruments RTL Toolset within the main menu system. Available options are shown below.

For complete instructions on the use of the RTL tools, please refer to the *TI VHSIC RTL Toolset User's Manual* (see Section 6). A sample design scenario on the use of the RTL Toolset is shown in Appendix C of the JRS Integrated Hardware Design Toolset User's Manual.

2.4.4 Converting Between VHDL and RDD

The VHDL/RDD conversion tools embedded in the workstation allow the user to convert VHDL behavioral descriptions into RDD descriptions for

Figure 2.2: RTL Toolset Main Menu

```
-----------------------------------------------------[rtl]
                RTL TOOLSET MAIN MENU
-----------------------------------------------------

          [S] SETUP USER's PROFILE
          [C] CHANGE RELEASE DIRECTORY

          [1] Makemodl        [2] Makenode
          [3] Modl            [4] Node
          [5] Preamble        [6] Tailor
          [7] Srtl            [8] Brtl
          [9] Crtl            [10] Linker
          [11] Slice          [12] Sim

          [H] HELP              [X] EXIT/RETURN

     Enter Selection ==>

-----------------------------------------------------
```

purposes of building a JRS fast simulator for the design. However, the VHDL descriptions must be in the form that the JRS Hardware Design Tool accepts so a VHDL Style Guide (see Section 6) is provided. The workstation also converts RDD machine designs that have been generated using the RTL tools or the RDD User Input program into behavioral VHDL descriptions.

2.4.5 Network Communications Summary

The JRS Integrated Hardware Design Toolset configuration contains the hardware and software necessary for the PC to appear as a network node. In addition, the workstation provides a user-friendly environment that assists designers in using VHDL and RTL. The PC is able to communicate with network nodes via

- TCP/IP.
- File transfer capability using FTP to transfer PC/VHDL files to the VAX.
- NFS.

This system provides considerable flexibility in that a designer can either log onto the PC and use a Sun or VAX as sharable system resources, or use the PC as a single user VHDL system. The PC also supports multiple VHDL users simultaneously performing VHDL and/or RTL functions because it uses the Unix operating system which provides multiple concurrent sessions.

In any mode of operation, the designer creates and analyzes VHDL descriptions, generates models, builds and executes simulators, and generates reports, all on the PC. RTL code can be written and executed on VHDL simulators (both low-level or behavioral), a JRS fast simulator, and/or the RTL interactive debugger simulator. All updated source files can be backed-up on the VAX or Sun disks and the network print server can be used for listings and reports. The PC-based workstation uses TCP/IP on the VAX and PC Unix communications software to accomplish the network tasks.

Unix Shell files were written to perform several network tasks that are available from the main menu. In summary, the tasks are

The functional description of each network menu selection can be found in Appendix A of the *JRS Integrated Hardware Design Toolset User's Manual* (refer to Section 6).

Figure 2.3: Network Communications Main Menu

```
-----------------------------------------------------[net]
          NETWORK COMMUNICATIONs MAIN MENU
-------------------------------------------------

          [1] SAVE FILES TO REMOTE MACHINE
          [2] RETRIEVE FILES FROM REMOTE MACHINE
          [3] SAVE DIRECTORY TO REMOTE MACHINE
          [4] RESTORE DIRECTORY FROM REMOTE MACHINE

          [H] HELP           [X] EXIT/RETURN

        Enter Selection ==>

-------------------------------------------------
```

Chapter 3

Deliverables

The object code for the JRS Integrated Hardware Design Toolset and a complete hardware workstation has been delivered to the government site at:

1. Fort Monmouth, Fred Glickman

Additional object code systems for the toolset were provided to:

1. Wright Aeronautical Laboratories, John Hines
2. Naval Research Laboratories, J.P. Letellier

The following VAX software is included in each delivery.

- Unixtalk.
- Command files necessary to transfer files from the VAX to the PC.

The following PC software is *not* included in this delivery but is necessary for the execution of the JRS Integrated Hardare Design Toolset.

- Interactive Unix Workstation Developer (it includes the C compiler and C library as well as network software).

The following PC hardware is *not* included in the software deliveries but is necessary for the execution of the PC/VHDL workstation.

- 386 PC with keyboard
- Memory expansion card
- 8 MB RAM
- CDC ST4702N SCSI 600 MB hard disk drive with CDC 3280C SCSI controller –or– 1 or 2 CDC 155 MB hard disk drives with ESDI controller
- TEAC 5.25" 1.2MB floppy disk drive
- VGA, EGA, or Hercules card and monitor
- 3COM Etherlink II 3C501

Chapter 4

Accomplishments

4.1 Satisfaction of Objectives

The primary goal of this SBIR effort was to provide a low-cost VHDL designer workstation. This goal was accomplished and the project was conducted on-schedule and within budget. The PC/VHDL workstation not only supports VHDL, but additional functionality was provided to allow the hardware designer to design individual components and to test execution using RTL code. Two new simulators were provided in addition to the VHDL structural and behavioral simulators to give the user an interactive debugger (the RTL simulator) and a fast behavioral simulator (JRS Fast Sim). Translators were added to convert VHDL descriptions into an RTL Design Database (RDD) form that is compatible with the RTL toolset. Additional translators convert RDD representations into VHDL.

The PC-based VHDL system consists of the Intermetrics Sun 3 Version 3.0 VHDL software ported to the PC. The workstation has additional hardware and software that allow the PC to appear as a network node. JRS also ported and integrated the Texas Instruments RTL subsystem software to provide the user with additional hardware and software design capabilities. The network capabilities are layered on top of the baseline VHDL version, augmenting its functions. The requirements were as follows:

1. All executable modules of the PC-based VHDL System shall be built in such a way so that they will take advantage of standardized system calls. The intent of the requirement is to produce a product that is portable in order to take advantage of the latest technologies in processing power and software, and to produce a product that has excellent performance

2. The user interface shall include a number of menu displays to support VHDL development.

3. Support the use of the PC/VHDL system by multiple users.

4. PC Communications Software: The vendor-supplied PC communications software, which contains all device drivers and software necessary to establish a logical link between the server and client, shall support the following activities from the PC server:

   (a) Initiate file transfer between client and PC server.

   (b) Access remote printers and other network resources.

   (c) Support task-to-task interface with multiple clients.

   (d) Allow the server disk resources to be utilized by client programs. Multiple clients may access the server disk but it is assumed that each user will have a different library structure.

   The client systems shall be any network node.

5. A test suite shall demonstrate that the above requirements have been met.

6. A user's manual shall describe completely the use of the network version of the PC-based VHDL System.

7. An automatic procedure shall install the PC-based VHDL System software. The procedure shall create directories, transfer files from floppy to hard disk, and may modify system files to set search paths, define environments, custom tailor system according to user hardware, and any other action necessary for automatic, one-command installation. The command procedure may query the installer for some information, and make recommendations or suggestions.

The above requirements have all been met and even exceeded because additional hardware design tools were integrated and included in the PC/VHDL system.

One original direction of the Phase I contract was to provide a minimum PC/VHDL capability that would run on 80286-based IBM PC/ATs (and clones). This effort was to concentrate on defining a spectrum of PC/AT configurations on which the PC-based VHDL System would run. Ideally, the PC-based VHDL System was to execute within the limitations of the 80286, and on systems with a rather limited hard disk system, such as 20 megabytes. It was not to handle particularly large VHDL designs, nor was it to execute particularly fast. A separate test suite was to be used on the 80286 version

to allow for smaller models. This direction was not possible due to the large amount of disk space needed by the PC/VHDL system, including the UNIX environment, runtime environment, and room for user directories and library space. Also, it appears that PC technology is progressing so fast that an 80286 version may not be desirable from a marketing perspective, especially in light of the extensive development work that would be necessary even if JRS could make an 80286 version with limited disk space. It was therefore recommended that this direction not be pursued.

4.2 Functional Testing

Testing of the IDAS was required to:

1. Verify that the PC/VHDL system was functionally equivalent to the Intermetrics Sun 3 VHDL system.
2. Verify that the RTL simulator and the JRS Fast Sim generated the same results as the VHDL structural and behavioral simulators.

The major source for obtaining VHDL test programs was the Intermetrics VHDL IEEE test suite of 704 programs. A new Unix shell was written to automatically call the tests and save the results. Disk space was a problem so subsets of the suite were transmitted to the PC/VHDL one at a time, executed, and disk space was cleared between subset executions. These VHDL programs were executed successfully on the PC/VHDL system and yielded identical answers to the Intermetrics Sun 3 execution of the same test suite. Appendix A shows the test results from executing the 704 Intermetrics VHDL programs. At the end of the appendix, 14 failures are reported but Intermetrics verified that the tests were in error and should be deleted from the test suite because the Sun 3 Version 2.1 generates the same errors. Four of the errors exist only in Version 2.0 and have since been corrected in Version 2.1.

JRS ran the Intermetrics-supplied Characterization Test Suite using the ported system on the PC. Elapsed time, CPU time, memory requirements, disk usage, and other significant metrics were recorded for the purposes of demonstrating the limits of the PC/VHDL system associated with characteristics of any specific installation configuration. Limitations do not seem to exist for the representative machine descriptions used for testing. Appendix D contains the Intermetrics study on characterization of VHDL features with regard to performance. Test data is included at the end of report to show the measurements for 31 VHDL test cases and the statistics for each VHDL feature tested.

JRS has demonstrated the features listed in Section 4.1 by entering machine descriptions for 3 separate processors: MACH5, Aladdin, and OBP. JRS then generated machine microcode for a subset of ACVC tests using JRS Ada compilers and executed that microcode on several different simulators for each machine. Appendix B shows execution timing to assemble, link, and simulate an ACVC test suite on the workstation RTL simulator. Appendix C shows those same ACVC tests executing on a Sun 4. Although the Sun 4 is much faster, the PC/VHDL speeds are respectable and much faster than a MicroVAX II.

The following list details the testing that was performed on the PC/VHDL workstation.

1. Input MACH5 VHDL machine description and perform VHDL Analyze, Model Generate, and Build Simulator.
2. Input Aladdin VHDL machine description and perform VHDL Analyze, Model Generate, and Build Simulator.
3. Input OBP VHDL machine description and perform VHDL Analyze, Model Generate, and Build Simulator.
4. Execute Aladdin machine code for 145 ACVC tests on the Aladdin VHDL behavioral simulator.
5. Execute Aladdin machine code for 3 TI benchmarks on the Aladdin VHDL behavioral simulator.
6. Execute OBP machine code for 99 ACVC tests on the OBP VHDL behavioral simulator.
7. Execute MACH5 machine code for 145 ACVC tests on the MACH5 VHDL behavioral simulator.
8. Use the RDD User Input program to generate a machine description of the MACH5 processor.
9. Execute 704 IEEE VHDL tests on the Intermetrics Sun 3 system as well as the PC/VHDL system. Verify that results are identical.
10. Build an RDD machine description of the Aladdin processor. Build an RTL simulator for it.
11. Execute 3 TI benchmarks on the Aladdin RTL simulator.
12. Execute 145 ACVC tests on the Aladdin RTL simulator using microcode that was generated by the JRS Aladdin Ada compiler.

13. Translate the Aladdin RDD description to VHDL and build a VHDL behavioral simulator for it. Verify that it is functionally equivalent to the VHDL behavioral simulator that was built directly from the VHDL model.

14. Execute 3 TI benchmarks on the Aladdin VHDL behavioral simulator using microcode that was generated by the JRS Aladdin Ada compiler.

15. Execute 145 ACVC tests on the Aladdin VHDL behavioral simulator using microcode that was generated by the JRS Aladdin Ada compiler.

16. Build a JRS Fast Sim from the Aladdin RDD description.

17. Execute 3 TI benchmarks on the Aladdin Fast Sim using microcode that was generated by the JRS Aladdin Ada compiler.

18. Execute 145 ACVC tests on the Aladdin Fast Sim using microcode that was generated by the JRS Aladdin Ada compiler.

19. Translate the MACH5 VHDL description to RDD and then translate it back to VHDL. Build a VHDL behavioral simulator for it. Verify that it is equivalent to the VHDL behavioral simulator generated directly from the VHDL description.

20. Execute 145 ACVC tests on the MACH5 VHDL behavioral simulator using microcode that was generated by the JRS MACH5 Ada compiler.

21. Build a JRS Fast Sim for the MACH5 processor from the RDD description.

22. Execute 145 ACVC tests on the MACH5 Fast Sim using microcode that was generated by the JRS MACH5 Ada compiler.

23. Translate the OBP VHDL description to RDD and build a VHDL behavioral simulator for it.

24. Execute 99 ACVC tests on the OBP VHDL behavioral simulator using microcode that was generated by the JRS OBP Ada compiler.

25. Build a JRS Fast Sim for the OBP processor from the RDD description.

26. Execute 99 ACVC tests on the OBP Fast Sim using microcode that was generated by the JRS OBP Ada compiler.

In addition to these functional tests, JRS also performed 5 communications tests to verify correct execution of the communications protocols available on the PC/VHDL system.

Chapter 5

Conclusions and Recommendations

5.1 Benefits

JRS Research Laboratories has successfully built a PC-based VHDL system that contains all of the functionality and speed of the Intermetrics Sun-3 Version 2.0 VHDL. The JRS PC/VHDL workstation contains:

1. User workstation that is menu-driven.
2. The complete Intermetrics IEEE 1076 VHDL Version 2.0 and all of the corresponding Sun 3 VHDL commands.
3. The complete RTL toolset including
   - A language for describing the RTL.
   - A language for describing the bus connections between individually programmable machines.
   - Tools to create an RTL assembler and linker.
   - Tools to create an RTL level simulator and debugger.
   - A tool to generate the bit settings for the microword used as input to the ROM.
4. The complete RTL-level Design Database (RDD) toolset including
   - Tools to input RDD machine descriptions.
   - Tools to convert RDD machine descriptions into VHDL descriptions.

- Tools to convert VHDL behavioral descriptions into RDD descriptions.
- Tools to generate VHDL behavioral simulators from RDD.
- Tools to generate Ada fast simulators from RDD.

5. Network communications to perform the following tasks:
   - Save files from PC to Vax.
   - Get files from Vax to PC.
   - Save complete PC directory on Vax.
   - Restore the complete saved directory from Vax to PC.
   - Save files from Vax to PC.
6. User commands to perform the above 4 sets of work.
7. Several complete user's guides describing the above capabilities.
8. Numerous examples, including a sample RUI input session for the MACH5 machine.
9. Test programs, test data, and test results from the following:
   - Executing IEEE VHDL tests on the PC/VHDL system
   - Generating a VHDL behavioral simulator for 2 different machines described in VHDL: Analytyx On-Board Processor (OBP) and MACH5, a synthesized processor based on silicon compiler primitives.
   - Generating a VHDL behavioral simulator for the 2 machines described in RDD.
   - Generating an Ada fast simulator for the 2 machines described in VHDL.
   - Generating an Ada fast simulator for the 2 machines described in RDD.
   - Executing JRS-compiled microcode for the 2 machines on the VHDL behavioral simulator.
   - Executing JRS-compiled microcode for the 2 machines on the Ada fast simulator.
10. The manual also includes a description of how to interface microprograms and data to the simulators generated by the PC/VHDL RDD/RTL system. JRS tested the interface by using symbol table information and microcode generated by two JRS Ada compiler systems and executing the load modules on simulators generated on the

PC/VHDL workstation. The load modules executed were for the OBP processor (an NRL machine) and the MACH5 processor (a design synthesized using the JRS IDAS system).

11. Intermetrics results from trying to characterize VHDL features and generate useful analysis information for estimating disk size and CPU time based on a given test suite.

5.2 Restrictions

JRS had hoped to port the PC/VHDL system to an 80286 processor running MS-DOS. Due to the size of the modules and the support environment they need, this port was not possible. However, it appears that PC technology is progressing so fast that an 80286 version would probably not be desirable from a marketing perspective. It was therefore recommended that this direction not be pursued.

5.3 Future Plans and Possible New Paths

Future plans for the JRS Integrated Hardware Design Toolset are unclear due to JRS rights associated with the VHDL source code that was written by Intermetrics. Valid purchased the rights and now Cadence has purchased the rights. For purposes of this final report, some thoughts are offered in this section as potential new paths, but with some logistics to be worked out.

1. It would be beneficial to implement the GNU C compiler with the PC/VHDL simulator generator. The PC/VHDL system currently uses the Interactive C compiler because of its complete runtime library. However, GNU is now coming out with a runtime library. Integrating PC/VHDL with GNU would allow a lower cost Unix environment for users.
2. X-Windows user interface.
3. Graphics input of designs.
4. Graphics output of designs.
5. Integration with additional VHDL-based tools such as network modeling, processor synthesis, and logic synthesis tools.

6. Analysis reports giving the user more insight into the simulation results.

7. Integration of the Intermetrics Characterization test suite, which generates statistics about individual VHDL features affecting disk space and CPU time for a given target machine, with the Intermetrics-supplied analysis stand-alone program that parses VHDL programs and outputs feature counts. This can be combined with computer statistics derived from the Characterization test suite report to help users tailor their VHDL code if a given PC host configuration is too limiting in disk space usage and CPU time for VHDL functions.

8. Market the JRS Integrated Hardware Design Toolset to universities and the commercial marketplace.

Chapter 6

Referenced Documents

Specifications:

- *IEEE Standard VHDL language Reference Manual*, IEEE Std 1076-1987, The Institute of Electrical and Electronics Engineers, Inc.
- *Technical Memorandum, Evaluation of Runtime Environments*, TM:DMH:0942, 6 December 1989, JRS Research Laboratories Inc.
- *Plan for Remaining Development for the PC Based VHDL Design System*, 2 Nov 1990,JRS Research Laboratories Inc.
- *JRS Integrated Hardware Design Toolset User's Manual*, SUM.0129-00, 22 January 1992, JRS Research Laboratories Inc.
- *PC/VHDL User's Manual*, SUM.0115-00, October 1991, JRS Research Laboratories Inc.
- *RDD User Input Manual*, SUM.0115-00, October 1991, JRS Research Laboratories Inc.
- *Texas Instruments VHSIC Register Transfer Language Toolset VAX/VMS User's Guide*, 4 March 1988, Texas Instruments, Inc.

Chapter 7

Acronyms

ACS	Ada Compiler System for UYS-2.
AIL	Ada Intermediate Language, the intermediate level program representation generated by the Ada compiler.
BRTL	Bit RTL - Describes the bit portion of the RTL (part of the Texas Instruments RTL Toolset).
CRTL	Code RTL - Describes the simulation code portion of the RTL (part of the Texas Instruments RTL Toolset).
IDAS	Integrated Design Automation System, a JRS product that synthesizes machine designs and generates Ada compiler environments for them.
GS	Generate Simulator - A tool that generates a "fast" simulator from any RDD machine description (part of the JRS VHDL Software Level Simulator Toolset).
GV	Generate VHDL - A tool that converts RDD machine descriptions into VHDL behavioral descriptions (part of the JRS VHDL Software Level Simulator Toolset).
MG	Model Generator - A tool that translates IVAN into C source code (part of the Intermetrics Standard VHDL 1076 Support Environment).
MODL	Micro-Operation Description Language - An extended hardware description language (part of the Texas Instruments RTL Toolset).
PROM	Programmable ROM.
RDD	RTL Description Database - A library containing the RDD machine description files (part of the JRS VHDL Software Level Simulator Toolset).

RG	Report Generator - A tool that generates reports from a file of signal histories (part of the Intermetrics Standard VHDL 1076 Support Environment).
RTL	Register Transfer Language.
ROM	Read Only Memory.
RUI	RTL User Input - A question–answer program that converts user's responses into an RDD machine description file (part of the JRS VHDL Software Level Simulator Toolset).
SRG	simulation Report Generator - A tool that reads memory dump files from simulation and generates a file containing variable names and their values (part of the JRS VHDL Software Level Simulator Toolset).
SRTL	Source RTL - Describes the source program portion of the RTL (part of the Texas Instruments RTL Toolset).
VHDL	VHSIC Hardware Description Language.
VHSIC	Very High Speed Integrated Circuit.
VLS	VHDL Library System - A tool that provides the user of the VHDL 1076 Software System with the commands needed to maintain the Design Database (part of the Intermetrics Standard VHDL 1076 Support Environment).
VLSI	Very Large Scale Integrated Circuit.
VRT	VHDL to RDD Translator - A tool that translates VHDL behavioral descriptions into RDD machine descriptions (part of the JRS VHDL Software Level Simulator Toolset).

Appendix A

Test Results of Intermetrics Test Suite

```
 1   ct00001.vhd
 2   ct00002.vhd
 3   ct00003.vhd
 4   ct00004.vhd
 5   ct00005.vhd
 6   ct00006.vhd
 7   ct00007.vhd
 8   ct00008.vhd
 9   ct00009.vhd
10   ct00010.vhd
11   ct00011.vhd
12   ct00012.vhd
13   ct00013.vhd
14   ct00014.vhd
15   ct00015.vhd
16   ct00016.vhd
17   ct00018.vhd
18   ct00019.vhd
19   ct00020.vhd
20   ct00021.vhd
21   ct00022.vhd
22   ct00023.vhd
23   ct00024.vhd
24   ct00025.vhd
25   ct00026.vhd
26   ct00027.vhd
27   ct00028.vhd
28   ct00029.vhd
29   ct00030.vhd
30   ct00031.vhd
31   ct00032.vhd
32   ct00033.vhd
33   ct00034.vhd
34   ct00035.vhd
35   ct00036.vhd
36   ct00037.vhd
37   ct00041.vhd
38   ct00042.vhd
39   ct00043.vhd
40   ct00044.vhd
41   ct00045.vhd
42   ct00046.vhd
43   ct00047.vhd
44   ct00048.vhd
45   ct00049.vhd
46   ct00050.vhd
47   ct00051.vhd
48   ct00052.vhd
49   ct00053.vhd
50   ct00054.vhd
51   ct00055.vhd
52   ct00056.vhd
53   ct00057.vhd
54   ct00058.vhd
55   ct00059.vhd
```

```
    56   ct00060.vhd
```



```
    57   ct00061.vhd
    58   ct00062.vhd
    59   ct00063.vhd
    60   ct00064.vhd
    61   ct00065.vhd
    62   ct00066.vhd
    63   ct00067.vhd
    64   ct00068.vhd
    65   ct00069.vhd
    66   ct00070.vhd
    67   ct00071.vhd
    68   ct00072.vhd
    69   ct00073.vhd
    70   ct00074.vhd
    71   ct00075.vhd
    72   ct00076.vhd
    73   ct00077.vhd
    74   ct00078.vhd
    75   ct00079.vhd
    76   ct00080.vhd
    77   ct00081.vhd
    78   ct00082.vhd
    79   ct00083.vhd
    80   ct00084.vhd
    81   ct00085.vhd
    82   ct00086.vhd
    83   ct00087.vhd
    84   ct00088.vhd
    85   ct00089.vhd
    86   ct00090.vhd
    87   ct00091.vhd
    88   ct00092.vhd
    89   ct00093.vhd
    90   ct00094.vhd
    91   ct00095.vhd
    92   ct00096.vhd
    93   ct00097.vhd
    94   ct00098.vhd
    95   ct00099.vhd
    96   ct00100.vhd
    97   ct00101.vhd
    98   ct00102.vhd
    99   ct00103.vhd
   100   ct00104.vhd
   101   ct00105.vhd
   102   ct00106.vhd
   103   ct00107.vhd
   104   ct00108.vhd
   105   ct00109.vhd
```

```
106   ct00111.vhd
107   ct00112.vhd
108   ct00113.vhd
109   ct00114.vhd
110   ct00115.vhd
111   ct00116.vhd
112   ct00117.vhd
```



```
113   ct00118.vhd
114   ct00119.vhd
115   ct00120.vhd
116   ct00121.vhd
117   ct00122.vhd
118   ct00123.vhd
119   ct00124.vhd
120   ct00125.vhd
121   ct00126.vhd
122   ct00127.vhd
123   ct00128.vhd
124   ct00129.vhd
125   ct00130.vhd
126   ct00131.vhd
127   ct00132.vhd
128   ct00133.vhd
129   ct00134.vhd
130   ct00135.vhd
131   ct00136.vhd
132   ct00137.vhd
133   ct00138.vhd
134   ct00139.vhd
135   ct00140.vhd
136   ct00141.vhd
137   ct00142.vhd
138   ct00143.vhd
139   ct00144.vhd
140   ct00145.vhd
141   ct00146.vhd
142   ct00147.vhd
143   ct00148.vhd
144   ct00149.vhd
145   ct00150.vhd
146   ct00151.vhd
147   ct00152.vhd
148   ct00153.vhd
149   ct00154.vhd
150   ct00155.vhd
151   ct00156.vhd
152   ct00157.vhd
153   ct00158.vhd
154   ct00159.vhd
155   ct00160.vhd
```

```
156   ct00161.vhd
157   ct00162.vhd
158   ct00163.vhd
159   ct00164.vhd
160   ct00165.vhd
161   ct00166.vhd
162   ct00167.vhd
163   ct00168.vhd
164   ct00169.vhd
165   ct00170.vhd
166   ct00171.vhd
167   ct00172.vhd
168   ct00173.vhd
```



```
169   ct00174.vhd
170   ct00175.vhd
171   ct00176.vhd
172   ct00177.vhd
173   ct00178.vhd
174   ct00179.vhd
175   ct00180.vhd
176   ct00181.vhd
177   ct00182.vhd
178   ct00183.vhd
179   ct00184.vhd
180   ct00185.vhd
181   ct00186.vhd
182   ct00187.vhd
183   ct00188.vhd
184   ct00189.vhd
185   ct00190.vhd
186   ct00191.vhd
187   ct00192.vhd
188   ct00193.vhd
189   ct00194.vhd
190   ct00195.vhd
191   ct00196.vhd
192   ct00197.vhd
193   ct00198.vhd
194   ct00199.vhd
195   ct00200.vhd
196   ct00201.vhd
197   ct00202.vhd
198   ct00203.vhd
199   ct00204.vhd
200   ct00205.vhd
201   ct00206.vhd
202   ct00207.vhd
203   ct00208.vhd
204   ct00209.vhd
205   ct00210.vhd
```

206 ct00211.vhd
207 ct00212.vhd
208 ct00213.vhd
209 ct00214.vhd
210 ct00215.vhd
211 ct00216.vhd
212 ct00217.vhd
213 ct00218.vhd
214 ct00219.vhd
215 ct00220.vhd
216 ct00221.vhd
217 ct00222.vhd
218 ct00223.vhd
219 ct00224.vhd
220 ct00225.vhd
221 ct00226.vhd
222 ct00227.vhd
223 ct00228.vhd
224 ct00229.vhd

225 ct00230.vhd
226 ct00231.vhd
227 ct00232.vhd
228 ct00233.vhd
229 ct00234.vhd
230 ct00235.vhd
231 ct00236.vhd
232 ct00237.vhd
233 ct00238.vhd
234 ct00243.vhd
235 ct00244.vhd
236 ct00245.vhd
237 ct00246.vhd
238 ct00247.vhd
239 ct00248.vhd
240 ct00249.vhd
241 ct00250.vhd
242 ct00251.vhd
243 ct00252.vhd
244 ct00254.vhd
245 ct00255.vhd
246 ct00256.vhd
247 ct00257.vhd
248 ct00258.vhd
249 ct00259.vhd
250 ct00260.vhd
251 ct00261.vhd
252 ct00262.vhd
253 ct00263.vhd
254 ct00264.vhd
255 ct00265.vhd

```
256   ct00266.vhd
257   ct00267.vhd
258   ct00268.vhd
259   ct00269.vhd
260   ct00270.vhd
261   ct00271.vhd
262   ct00272.vhd
263   ct00273.vhd
264   ct00274.vhd
265   ct00275.vhd
266   ct00276.vhd
267   ct00277.vhd
268   ct00278.vhd
269   ct00279.vhd
270   ct00280.vhd
271   ct00281.vhd
272   ct00282.vhd
273   ct00283.vhd
274   ct00284.vhd
275   ct00285.vhd
276   ct00286.vhd
277   ct00287.vhd
278   ct00288.vhd
279   ct00289.vhd
280   ct00290.vhd
```



```
281   ct00291.vhd
282   ct00292.vhd
283   ct00293.vhd
284   ct00294.vhd
285   ct00295.vhd
286   ct00296.vhd
287   ct00297.vhd
288   ct00298.vhd
289   ct00300.vhd
290   ct00301.vhd
291   ct00302.vhd
292   ct00303.vhd
293   ct00304.vhd
294   ct00305.vhd
295   ct00306.vhd
296   ct00307.vhd
297   ct00308.vhd
298   ct00309.vhd
299   ct00310.vhd
300   ct00311.vhd
301   ct00312.vhd
302   ct00313.vhd
303   ct00314.vhd
304   ct00315.vhd
305   ct00316.vhd
```

```
306    ct00318.vhd
307    ct00319.vhd
308    ct00320.vhd
309    ct00321.vhd
310    ct00322.vhd
311    ct00323.vhd
312    ct00324.vhd
313    ct00325.vhd
314    ct00326.vhd
315    ct00327.vhd
316    ct00328.vhd
317    ct00329.vhd
318    ct00330.vhd
319    ct00331.vhd
320    ct00332.vhd
321    ct00333.vhd
322    ct00334.vhd
323    ct00335.vhd
324    ct00336.vhd
325    ct00337.vhd
326    ct00338.vhd
327    ct00339.vhd
328    ct00340.vhd
329    ct00341.vhd
330    ct00342.vhd
331    ct00343.vhd
332    ct00344.vhd
333    ct00345.vhd
334    ct00346.vhd
335    ct00347.vhd
336    ct00348.vhd
```



```
337    ct00349.vhd
338    ct00350.vhd
339    ct00351.vhd
340    ct00352.vhd
341    ct00353.vhd
342    ct00354.vhd
343    ct00355.vhd
344    ct00356.vhd
345    ct00357.vhd
346    ct00358.vhd
347    ct00359.vhd
348    ct00360.vhd
349    ct00361.vhd
350    ct00362.vhd
351    ct00363.vhd
352    ct00364.vhd
353    ct00365.vhd
354    ct00366.vhd
355    ct00367.vhd
```

```
356   ct00368.vhd
357   ct00369.vhd
358   ct00370.vhd
359   ct00371.vhd
360   ct00372.vhd
361   ct00373.vhd
362   ct00374.vhd
363   ct00375.vhd
364   ct00376.vhd
365   ct00377.vhd
366   ct00378.vhd
367   ct00379.vhd
368   ct00380.vhd
369   ct00381.vhd
370   ct00382.vhd
371   ct00383.vhd
372   ct00384.vhd
373   ct00385.vhd
374   ct00386.vhd
375   ct00387.vhd
376   ct00388.vhd
377   ct00389.vhd
378   ct00390.vhd
379   ct00391.vhd
380   ct00392.vhd
381   ct00393.vhd
382   ct00394.vhd
383   ct00395.vhd
384   ct00396.vhd
385   ct00397.vhd
386   ct00398.vhd
387   ct00399.vhd
388   ct00400.vhd
389   ct00401.vhd
390   ct00402.vhd
391   ct00403.vhd
392   ct00404.vhd
```



```
393   ct00405.vhd
394   ct00406.vhd
395   ct00407.vhd
396   ct00408.vhd
397   ct00409.vhd
398   ct00410.vhd
399   ct00411.vhd
400   ct00412.vhd
401   ct00413.vhd
402   ct00414.vhd
403   ct00415.vhd
404   ct00416.vhd
405   ct00417.vhd
```

406 ct00418.vhd
407 ct00419.vhd
408 ct00420.vhd
409 ct00421.vhd
410 ct00422.vhd
411 ct00423.vhd
412 ct00424.vhd
413 ct00425.vhd
414 ct00426.vhd
415 ct00427.vhd
416 ct00428.vhd
417 ct00429.vhd
418 ct00430.vhd
419 ct00431.vhd
420 ct00432.vhd
421 ct00433.vhd
422 ct00434.vhd
423 ct00435.vhd
424 ct00436.vhd
425 ct00437.vhd
426 ct00438.vhd
427 ct00439.vhd
428 ct00440.vhd
429 ct00441.vhd
430 ct00442.vhd
431 ct00443.vhd
432 ct00444.vhd
433 ct00445.vhd
434 ct00447.vhd
435 ct00448.vhd
436 ct00449.vhd
437 ct00451.vhd
438 ct00452.vhd
439 ct00453.vhd
440 ct00455.vhd
441 ct00457.vhd
442 ct00458.vhd
443 ct00459.vhd
444 ct00460.vhd
445 ct00461.vhd
446 ct00462.vhd
447 ct00463.vhd
448 ct00464.vhd

449 ct00465.vhd
450 ct00466.vhd
451 ct00467.vhd
452 ct00468.vhd
453 ct00469.vhd
454 ct00470.vhd
455 ct00471.vhd

```
456   ct00472.vhd
457   ct00473.vhd
458   ct00474.vhd
459   ct00475.vhd
460   ct00476.vhd
461   ct00477.vhd
462   ct00478.vhd
463   ct00479.vhd
464   ct00480.vhd
465   ct00481.vhd
466   ct00482.vhd
467   ct00483.vhd
468   ct00484.vhd
469   ct00485.vhd
470   ct00486.vhd
471   ct00487.vhd
472   ct00488.vhd
473   ct00489.vhd
474   ct00490.vhd
475   ct00491.vhd
476   ct00492.vhd
477   ct00493.vhd
478   ct00494.vhd
479   ct00495.vhd
480   ct00496.vhd
481   ct00497.vhd
482   ct00498.vhd
483   ct00499.vhd
484   ct00500.vhd
485   ct00501.vhd
486   ct00503.vhd
487   ct00505.vhd
488   ct00506.vhd
489   ct00507.vhd
490   ct00508.vhd
491   ct00509.vhd
492   ct00510.vhd
493   ct00511.vhd
494   ct00512.vhd
495   ct00513.vhd
496   ct00514.vhd
497   ct00515.vhd
498   ct00517.vhd
499   ct00518.vhd
500   ct00519.vhd
501   ct00520.vhd
502   ct00521.vhd
503   ct00522.vhd
504   ct00523.vhd
```



```
505   ct00524.vhd
```

506 ct00525.vhd
507 ct00526.vhd
508 ct00527.vhd
509 ct00528.vhd
510 ct00529.vhd
511 ct00530.vhd
512 ct00531.vhd
513 ct00532.vhd
514 ct00533.vhd
515 ct00534.vhd
516 ct00535.vhd
517 ct00536.vhd
518 ct00537.vhd
519 ct00538.vhd
520 ct00539.vhd
521 ct00540.vhd
522 ct00541.vhd
523 ct00542.vhd
524 ct00543.vhd
525 ct00544.vhd
526 ct00545.vhd
527 ct00546.vhd
528 ct00547.vhd
529 ct00548.vhd
530 ct00549.vhd
531 ct00550.vhd
532 ct00551.vhd
533 ct00552.vhd
534 ct00553.vhd
535 ct00554.vhd
536 ct00555.vhd
537 ct00556.vhd
538 ct00557.vhd
539 ct00558.vhd
540 ct00559.vhd
541 ct00560.vhd
542 ct00561.vhd
543 ct00562.vhd
544 ct00563.vhd
545 ct00564.vhd
546 ct00565.vhd
547 ct00566.vhd
548 ct00567.vhd
549 ct00568.vhd
550 ct00569.vhd
551 ct00570.vhd
552 ct00571.vhd
553 ct00573.vhd
554 ct00574.vhd
555 ct00575.vhd
556 ct00576.vhd
557 ct00577.vhd
558 ct00578.vhd
559 ct00579.vhd
560 ct00580.vhd

```
561   ct00581.vhd
562   ct00582.vhd
563   ct00583.vhd
564   ct00584.vhd
565   ct00585.vhd
566   ct00586.vhd
567   ct00587.vhd
568   ct00588.vhd
569   ct00589.vhd
570   ct00590.vhd
571   ct00591.vhd
572   ct00592.vhd
573   ct00593.vhd
574   ct00594.vhd
575   ct00595.vhd
576   ct00596.vhd
577   ct00597.vhd
578   ct00598.vhd
579   ct00599.vhd
580   ct00600.vhd
581   ct00601.vhd
582   ct00603.vhd
583   ct00604.vhd
584   ct00605.vhd
585   ct00606.vhd
586   ct00607.vhd
587   ct00608.vhd
588   ct00609.vhd
589   ct00610.vhd
590   ct00611.vhd
591   ct00612.vhd
592   ct00613.vhd
593   ct00614.vhd
594   ct00615.vhd
595   ct00616.vhd
596   ct00617.vhd
597   ct00618.vhd
598   ct00619.vhd
599   ct00620.vhd
600   ct00621.vhd
601   ct00622.vhd
602   ct00623.vhd
603   ct00624.vhd
604   ct00625.vhd
605   ct00626.vhd
606   ct00627.vhd
607   ct00628.vhd
608   ct00629.vhd
609   ct00630.vhd
610   ct00631.vhd
611   ct00632.vhd
612   ct00633.vhd
613   ct00634.vhd
614   ct00635.vhd
615   ct00636.vhd
```

616 ct00637.vhd

617 ct00638.vhd
618 ct00639.vhd
619 ct00640.vhd
620 ct00641.vhd
621 ct00642.vhd
622 ct00643.vhd
623 ct00644.vhd
624 ct00645.vhd
625 ct00646.vhd
626 ct00647.vhd
627 ct00648.vhd
628 ct00649.vhd
629 ct00651.vhd
630 ct00652.vhd
631 ct00653.vhd
632 ct00654.vhd
633 ct00655.vhd
634 ct00657.vhd
635 ct00658.vhd
636 ct00659.vhd
637 ct00660.vhd
638 ct00661.vhd
639 ct00662.vhd
640 ct00665.vhd
641 ct00666.vhd
642 ct00667.vhd
643 ct00668.vhd
644 ct00669.vhd
645 ct00670.vhd
646 ct00671.vhd
647 ct00672.vhd
648 ct00673.vhd
649 ct00674.vhd
650 ct00677.vhd
651 ct00678.vhd
652 ct00679.vhd
653 ct00680.vhd
654 ct00681.vhd
655 ct00682.vhd
656 ct00683.vhd
657 ct00684.vhd
658 ct00685.vhd
659 ct00686.vhd
660 ct00687.vhd
661 ct00688.vhd
662 ct00689.vhd
663 ct00690.vhd
664 ct00691.vhd
665 ct00692.vhd

```
  666   ct00693.vhd
  667   ct00694.vhd
  668   ct00695.vhd
  669   ct00696.vhd
  670   ct00697.vhd
  671   ct00698.vhd
  672   ct00699.vhd
```



```
  673   ct00700.vhd
  674   ct00701.vhd
  675   ct00702.vhd
  676   ct00703.vhd
```

Jul 1 13:17 1990 results_bad Page 1

```
 1   ct00017.vhd : ENT00017(ARCH00017) : during mg
 2   ct00040.vhd : during vhdl
 3   ct00239.vhd : ENT00239(ARCH00239) : during mg
 4   ct00240.vhd : ENT00240(ARCH00240) : during mg
 5   ct00241.vhd : GENERIC_STANDARD_TYPES(ARCH00241) : during mg
 6   ct00242.vhd : GENERIC_STANDARD_TYPES(ARCH00242) : during mg
 7   ct00253.vhd : ENT00253(ARCH00253) : during mg
 8   ct00299.vhd : during vhdl
 9   ct00454.vhd : during vhdl
10   ct00456.vhd : during vhdl
11   ct00516.vhd : ENT00516_Test_Bench(ARCH00516_Test_Bench) : during mg
12   ct00663.vhd : E00000(ARCH00663) : during mg
13   ct00664.vhd : ENT00664(ARCH00664) : during mg
14   ct00704.vhd : ENT00704(ARCH00704) : during mg
```

From: IN%"dunlop@fanny.wash.inmet.COM" 26-NOV-1991 15:55:33.70
To: david@JRS.COM, debbie@JRS.COM
CC:
Subj: Re: VHDL IEEE Tests

Received: from ics.uci.edu by JRS.COM; Tue, 26 Nov 91 15:32 PST
Received: from inmet.camb.inmet.com by ics.uci.edu id aa07540; 26 Nov 91 11:38 PST
Received: from fanny.wash.inmet.com by inmet.camb.inmet.com (4.0/camb.inmet.com) id AA20160; Tue, 26 Nov 91 14:03:26 EST
Received: from sparky.wash.inmet.com by fanny.wash.inmet.com (4.1/SMI-4.1) id AA23543; Tue, 26 Nov 91 14:01:53 EST
Date: Tue, 26 Nov 91 14:01:53 EST
From: dunlop@fanny.wash.inmet.COM
Subject: Re: VHDL IEEE Tests
To: david@JRS.COM, debbie@JRS.COM
Cc: stearman@fanny.wash.inmet.COM
Message-Id: <9111261901.AA23543@fanny.wash.inmet.com>

> ct00017.vhd : ENT00017(ARCH00017) : during mg
> ct00239.vhd : ENT00239(ARCH00239) : during mg
> ct00240.vhd : ENT00240(ARCH00240) : during mg
> ct00241.vhd : GENERIC_STANDARD_TYPES(ARCH00241) : during mg
> ct00242.vhd : GENERIC_STANDARD_TYPES(ARCH00242) : during mg
> ct00253.vhd : ENT00253(ARCH00253) : during mg
> ct00299.vhd : during vhdl
> ct00516.vhd : ENT00516_Test_Bench(ARCH00516_Test_Bench) : during mg
> ct00704.vhd : ENT00704(ARCH00704) : during mg
>
> Please confirm that these 9 tests have been deleted from the IEEE
> VHDL test suite.

To the best of my knowledge these tests have never passed. Some of them (17, 239, 240, 241, 242, 253, 704) involve linkage ports which we do not support. 299 does not simulate due to the way our scripts treat warnings emitted from the Analyzer. Finally, 516 does not simulate because the model to be simulated is not complete (i.e., not simulatable).

> Are these known to have failed on the Sun-3 VHDL 2.0?
>
> ct00017.vhd : ENT00017(ARCH00017) : during mg
> ct00239.vhd : ENT00239(ARCH00239) : during mg
> ct00240.vhd : ENT00240(ARCH00240) : during mg
> ct00241.vhd : GENERIC_STANDARD_TYPES(ARCH00241) : during mg
> ct00242.vhd : GENERIC_STANDARD_TYPES(ARCH00242) : during mg
> ct00253.vhd : ENT00253(ARCH00253) : during mg
> ct00299.vhd : during vhdl
> ct00516.vhd : ENT00516_Test_Bench(ARCH00516_Test_Bench) : during mg
> ct00663.vhd : E00000(ARCH00663) : during mg
> ct00664.vhd : ENT00664(ARCH00664) : during mg
> ct00704.vhd : ENT00704(ARCH00704) : during mg

Concerning the two additional tests (663 & 664), 663 appears to fail but the root of this apparent problem is a language dispute wrt the 'STRUCTURE attribute. Test 664 is another test containing linkage ports.

Please let me know if I can be of more assistance.

Regards,

Doug Dunlop

Appendix B

ACVC Test Results on PC

*** PC/UNIX RTL ***

Assembly: SRTL/BRTL/CRTL

Program Name	Time
a22006b.rtl	20:54.8
a22006c.rtl	17:45.2
a22006d.rtl	18:12.0
a22006e.rtl	17:58.9
a22006f.rtl	17:43.7
a29002a.rtl	17:44.8
a29002b.rtl	17:50.8
a29002c.rtl	17:20.1
a29002d.rtl	17:07.3
a29002e.rtl	18:05.9
a29002f.rtl	17:15.5
a29002g.rtl	17:07.7
a29002h.rtl	17:38.0
a29002i.rtl	16:54.0
a29002j.rtl	16:51.0
a29003a.rtl	42:38.3
aa5008x.rtl	17:50.3
be_0001.rtl	6:37.0
be_0002.rtl	6:34.3
be_0003.rtl	10:31.9
be_0004.rtl	6:37.5
be_0005.rtl	6:34.8
be_0006.rtl	6:25.2
be_0007.rtl	6:38.8
be_0008.rtl	29:42.6
be_0009.rtl	23:23.9
be_000a.rtl	6:25.5
be_000b.rtl	27:26.0
be_000g.rtl	17:41.2
be_000i.rtl	15:44.8
be_000m.rtl	11:09.8
be_000y.rtl	6:38.0
c23001a.rtl	16:35.3
c23006a.rtl	16:42.0
c23006b.rtl	25:53.9
c23006c.rtl	27:22.8
c24002a.rtl	16:48.2
c24002b.rtl	17:24.2
c24003a.rtl	17:19.5
c24003b.rtl	16:26.9
c24113a.rtl	46:44.6
c24113b.rtl	45:01.7
c24113c.rtl	48:02.5
c24202a.rtl	16:42.1
c24202b.rtl	17:26.9
c24203a.rtl	17:16.0
c24207a.rtl	16:21.3
c25001a.rtl	17:15.8
c25001b.rtl	17:38.9
c25004a.rtl	18:29.1
c27001a.rtl	17:09.7
c27002a.rtl	17:29.6
c2a001a.rtl	17:29.3

c2a001b.rtl	16:13.5
c2a002a.rtl	17:04.9
c2a006a.rtl	16:58.9
c2a008a.rtl	17:35.6
c2a009a.rtl	17:34.3
c32203a.rtl	17:29.4
c34015b.rtl	25:01.6
c35106a.rtl	16:50.5
c35502g.rtl	37:48.1
c35503o.rtl	41:50.7
c35507o.rtl	29:52.2
c35508o.rtl	40:43.9
c35705a.rtl	37:45.6
c35705b.rtl	37:50.7
c35705c.rtl	37:17.5
c36203a.rtl	24:58.2
c45101a.rtl	1:10:21.4
c45101b.rtl	50:28.4
c45101c.rtl	37:16.9
c45101e.rtl	17:21.0
c45101g.rtl	5:35:48.4
c45101h.rtl	2:00:24.5
c45101i.rtl	56:48.6
c45101k.rtl	2:10:05.7
c45104a.rtl	17:50.9
c45122b.rtl	1:06:48.8
c45122c.rtl	1:15:59.9
c45123a.rtl	33:11.0
c45123b.rtl	36:44.1
c45123c.rtl	49:47.5
c45201a.rtl	1:30:13.9
c45201b.rtl	1:28:15.4
c45202a.rtl	23:17.0
c45202b.rtl	22:38.8
c45210a.rtl	1:03:35.7
c45211a.rtl	18:53.0
c45220a.rtl	1:28:43.8
c45220b.rtl	2:13:53.3
c45220c.rtl	2:22:56.4
c45220d.rtl	4:49:00.3
c45220e.rtl	17:24.3
c45220f.rtl	17:13.3
c45641a.rtl	56:43.1
c45641b.rtl	56:11.6
c45641c.rtl	56:00.2
c45662a.rtl	1:00:54.1
c45662b.rtl	1:37:26.7
c46011a.rtl	52:49.3
c49020a.rtl	25:05.2
c49021a.rtl	17:47.2
c52001b.rtl	34:07.2
c53004b.rtl	23:50.0
c53005a.rtl	16:31.8
c53005b.rtl	1:46:41.9
c53006a.rtl	18:32.4
c53006b.rtl	2:30:41.8
c53007a.rtl	1:20:07.0
c53008a.rtl	1:12:40.5
c54a03a.rtl	39:38.1
c54a23a.rtl	27:00.8

c54a26a.rtl	49:35.7
c54a27a.rtl	25:47.2
c54a41a.rtl	52:24.5
c54a42a.rtl	2:13:32.5
c54a42b.rtl	3:40:06.6
c54a42c.rtl	1:56:17.2
c54a42e.rtl	1:54:37.7
c54a42f.rtl	2:00:18.5
c55b03a.rtl	41:19.3
c55b04a.rtl	32:24.0
c55b06a.rtl	7:40:20.2
c55b06b.rtl	1:37:53.2
c55c01a.rtl	40:28.1
c55c02a.rtl	25:13.0
c55c03a.rtl	21:16.1
c55c03b.rtl	2:37:06.3
c56002a.rtl	2:25:44.8
c57002a.rtl	1:02:05.0
c57004a.rtl	33:48.4
c57004b.rtl	1:05:02.1
c57004c.rtl	43:32.0
c57005a.rtl	42:59.3
c64202a.rtl	1:01:53.2
c67003c.rtl	1:16:40.0
c74211b.rtl	56:28.2
c74402b.rtl	41:18.3
c83b02a.rtl	21:11.8
c83b02b.rtl	28:33.7
c83e03a.rtl	50:20.4
c83e04a.rtl	1:04:42.7
c85019a.rtl	20:58.2
c87b04c.rtl	21:16.9
c87b15a.rtl	20:50.3
c87b50a.rtl	23:22.9
ca1003a.rtl	41:49.3
ca1007a1m.rtl	27:54.4
ca1022a6m.rtl	34:52.4
ca1105a1m.rtl	27:54.7
ca3002a2m.rtl	29:50.7
ca3004e4m.rtl	54:01.7
ca3004f4m.rtl	1:00:31.5
ca3006c5m.rtl	29:30.8
ca3006d3m.rtl	27:03.7
cb1002a.rtl	23:19.2
cc1010a.rtl	27:31.6
fail_4660.rtl	7:55.8
flt.rtl	21:45.1
jfb.rtl	41:19.9
jrs0016.rtl	32:49.6
jrs0021.rtl	1:33:59.2
jrs0022.rtl	58:23.3
jrs0031.rtl	41:31.1
jrs0032.rtl	43:57.5
jrs0050.rtl	49:12.4
jrs0054.rtl	16:38.7
jrs0055.rtl	11:40.3
jrs0060.rtl	55:24.9
jrs0070.rtl	1:02:22.5
jrs0075.rtl	37:26.1
jrs0076.rtl	57:30.2

```
jrs0077.rtl                              36:26.1
jrs0078.rtl                              24:57.9
jrs0079.rtl                              24:42.6
jrs0080.rtl                            1:05:48.2
jrs0096.rtl                              30:12.8
jrs0210.rtl                              13:11.2
jrs0211.rtl                              13:44.0
jrs0240.rtl                            1:35:44.7
jrs0362.rtl                              57:47.5
jrs0710.rtl                              22:28.2
jrs0800.rtl                              17:20.3
jrs0802.rtl                              22:29.2
jrs0803.rtl                              18:26.5
jrs0810.rtl                              39:48.9
jrs0900.rtl                              28:06.3
jrs0901.rtl                              25:23.9*** ERROR ***    Error also occurs on
jrs0902.rtl                              28:21.2                 TI VAX system, so this
jrs0903.rtl                              37:27 7                 is a correct result.
jrs0905.rtl                              29:56.9
jrs0906.rtl                              23:24.2
jrsi830.rtl                              22:52.9
jrsi833.rtl                              23:31.4
jrsi836.rtl                              23:06.1
jrsi860.rtl                              45:50.0
jrsi861.rtl                              48:49.3
jrsi906.rtl                              21:49.3
qlt.rtl                                  24:50.8
rt_labels.rtl                            18:58.7
rt_not.rtl                               16:37.1
simple.rtl                                7:17.7
-------------------------------------------------
Number of tests: 203
```

*** PC/UNIX RTL ***

Linking: CRTL

Program Name	Time
a22006b	3:01.1
a22006c	3:10.3
a22006d	3:10.5
a22006e	3:09.4
a22006f	3:10.1
a29002a	3:09.4
a29002b	3:09.3
a29002c	3:09.8
a29002d	3:10.0
a29002e	3:09.7
a29002f	3:09.4
a29002g	3:08.9
a29002h	3:09.4
a29002i	3:10.1
a29002j	3:09.2
a29003a	3:26.0
aa5008x	3:10.6
be_0001	3:06.5
be_0002	3:06.3
be_0003	3:06.0
be_0004	3:06.3
be_0005	3:06.7
be_0006	3:06.8
be_0007	3:05.9
be_0008	3:13.5
be_0009	3:10.8
be_000a	3:06.1
be_000b	3:12.9
be_000g	3:09.2
be_000i	3:07.7
be_000m	3:08.4
be_000y	3:05.5
c23001a	3:09.9
c23006a	3:08.9
c23006b	3:13.3
c23006c	3:16.4
c24002a	3:09.4
c24002b	3:09.7
c24003a	3:10.3
c24003b	3:10.4
c24113a	3:20.6
c24113b	3:20.5
c24113c	3:20.9
c24202a	3:09.0
c24202b	3:10.1
c24203a	3:10.7
c24207a	3:10.8
c25001a	3:09.2
c25001b	3:10.2
c25004a	3:12.0
c27001a	3:09.9
c27002a	3:10.2
c2a001a	3:10.3

c2a001b	3:09.5
c2a002a	3:10.1
c2a006a	3:10.2
c2a008a	3:10.9
c2a009a	3:09.8
c32203a	3:11.0
c34015b	3:16.3
c35106a	3:10.4
c35502g	3:19.0
c35503o	3:22.4
c35507o	3:17.8
c35508o	3:21.8
c35705a	3:16.9
c35705b	3:17.1
c35705c	3:17.6
c36203a	3:14.4
c45101a	3:30.2
c45101b	3:21.9
c45101c	3:18.1
c45101e	3:09.9
c45101g	4:08.8
c45101h	3:42.3
c45101i	3:22.5
c45101k	3:46.9
c45104a	3:10.2
c45122b	3:04.5
c45122c	3:31.4
c45123a	3:15.6
c45123b	3:16.9
c45123c	3:23.7
c45201a	3:28.9
c45201b	3:30.8
c45202a	3:08.5
c45202b	3:09.5
c45210a	3:22.8
c45211a	3:09.8
c45220a	3:31.0
c45220b	3:46.9
c45220c	3:48.5
c45220d	5:13.8
c45220e	3:09.5
c45220f	3:10.1
c45641a	3:22.3
c45641b	3:22.3
c45641c	3:23.0
c45662a	3:24.1
c45662b	3:35.3
c46011a	3:25.6
c49020a	3:13.3
c49021a	3:09.1
c52001b	3:15.7
c53004b	3:11.9
c53005a	3:09.4
c53005b	3:36.1
c53006a	3:09.9
c53006b	3:52.3
c53007a	3:28.9
c53008a	3:28.4
c54a03a	3:17.7
c54a23a	3:12.8

c54a26a	3:20.1
c54a27a	3:13.0
c54a41a	3:21.8
c54a42a	3:48.0
c54a42b	4:06.2
c54a42c	3:40.0
c54a42e	3:41.5
c54a42f	3:43.8
c55b03a	3:16.9
c55b04a	3:14.7
c55b06a	5:24.8
c55b06b	3:32.1
c55c01a	3:16.6
c55c02a	3:12.3
c55c03a	3:11.0
c55c03b	3:46.4
c56002a	3:43.6
c57002a	3:23.2
c57004a	3:15.4
c57004b	3:25.5
c57004c	3:16.1
c57005a	3:16.1
c64202a	3:20.8
c67003c	3:29.3
c74211b	3:25.7
c74402b	3:18.9
c83b02a	3:09.1
c83b02b	3:14.5
c83e03a	3:02.2
c83e04a	3:22.5
c85019a	3:10.4
c87b04c	3:11.1
c87b15a	3:09.8
c87b50a	3:11.3
ca1003a	3:17.7
ca1007a1m	3:11.5
ca1022a6m	3:15.7
ca1105a1m	3:12.1
ca3002a2m	3:13.2
ca3004e4m	3:20.0
ca3004f4m	3:20.7
ca3006c5m	3:11.7
ca3006d3m	3:12.5
cb1002a	3:09.8
cc1010a	3:11.5
fail_4660	3:07.4
flt	3:10.0
jfb	3:02.2
jrs0016	3:14.6
jrs0021	3:29.3
jrs0022	3:21.6
jrs0031	3:16.9
jrs0032	3:17.4
jrs0050	3:17.9
jrs0054	3:09.6
jrs0055	3:07.1
jrs0060	3:21.9
jrs0070	3:23.7
jrs0075	3:17.5
jrs0076	3:20.7

```
jrs0077                                  3:14.4
jrs0078                                  3:10.8
jrs0079                                  3:12.5
jrs0080                                  3:22.2
jrs0096                                  3:14.4
jrs0210                                  3:07.2
jrs0211                                  3:06.2
jrs0240                                  3:29.7
jrs0362                                  3:21.9
jrs0710                                  3:11.0
jrs0800                                  3:09.6
jrs0802                                  3:10.0
jrs0803                                  3:09.0
jrs0810                                  3:16.3
jrs0900                                  3:12.2
jrs0901                                  3:11.2
jrs0902                                  3:11.6
jrs0903                                  3:16.9
jrs0905                                  3:16.1
jrs0906                                  3:11.3
jrsi830                                  3:11.4
jrsi906                                  3:09.7
qlt                                      3:10.6
rt_labels                                3:08.8
rt_not                                   3:08.8
simple                                   3:05.7
-----------------------------------------------
Number of tests: 199
```

*** PC/UNIX RTL ***
Simulation

Program Name	Cycles	Time (in sec)	Cycs/sec
a22006b	58	4.4	13.1818
a22006c	58	4.5	12.8889
a22006d	58	4.5	12.8889
a22006e	58	4.6	12.6087
a22006f	58	4.6	12.6087
a29002a	58	4.3	13.4884
a29002b	58	4.5	12.8889
a29002c	58	4.5	12.8889
a29002d	58	4.6	12.6087
a29002e	58	4.5	12.8889
a29002f	58	4.5	12.8889
a29002g	58	4.5	12.8889
a29002h	58	4.5	12.8889
a29002i	58	4.4	13.1818
a29002j	58	4.4	13.1818
a29003a	226	7.7	29.3506
aa5008x	58	4.5	12.8889
be_0001	28	3.8	7.36842
be_0002	28	4	7
be_0003	46	4.2	10.9524
be_0004	28	3.9	7.17949
be_0005	28	3.8	7.36842
be_0006	28	3.9	7.17949
be_0007	28	3.8	7.36842
be_0008	323	9.6	33.6458
be_0009	101	5.4	18.7037
be_000a	28	4	7
be_000b	118	5.7	20.7018
be_000g	646	15.9	40.6289
be_000i	315	9.4	33.5106
be_000m	51	4.4	11.5909
be_000y	28	3.9	7.17949
c23001a	58	4.6	12.6087
c23006a	58	4.4	13.1818
c23006b	91	5.2	17.5
c23006c	101	5.3	19.0566
c24002a	58	4.5	12.8889
c24002b	58	4.5	12.8889
c24003a	58	4.5	12.8889
c24003b	58	4.4	13.1818
c24113a	182	6.9	26.3768
c24113b	182	6.9	26.3768
c24113c	182	6.9	26.3768
c24202a	58	4.4	13.1818
c24202b	58	4.4	13.1818
c24203a	58	4.5	12.8889
c24207a	58	4.5	12.8889
c25001a	58	4.5	12.8889
c25001b	58	4.5	12.8889
c25004a	62	4.6	13.4783
c27001a	58	4.5	12.8889
c27002a	58	4.5	12.8889
c2a001a	58	4.4	13.1818
c2a001b	58	4.5	12.8889

c2a002a	58	4.5	12.8889
c2a006a	58	4.5	12.8889
c2a008a	58	4.5	12.8889
c2a009a	58	4.5	12.8889
c32203a	58	4.5	12.8889
c34015b	89	5	17.8
c35106a	58	4.4	13.1818
c35502g	381	10.8	35.2778
c35503o	158	6.4	24.6875
c35507o	106	5.4	19.6296
c35508o	153	6.3	24.2857
c35705a	152	6.4	23.75
c35705b	152	6.3	24.127
c35705c	152	6.3	24.127
c36203a	88	5	17.6
c45101a	1404	30.3	46.3366
c45101b	710	17.1	41.5205
c45101c	271	8.6	31.5116
c45101e	58	4.6	12.6087
c45101g	4938	98.4	50.1829
c45101h	2222	46.3	47.9914
c45101i	519	13.5	38.4444
c45101k	2275	47.3	48.0972
c45104a	58	4.4	13.1818
c45122b	1194	26.7	44.7191
c45122c	440	12	36.6667
c45123a	264	8.5	31.0588
c45123b	271	8.7	31.1494
c45123c	175	6.8	25.7353
c45201a	11616	226.1	51.3755
c45201b	9774	190.9	51.1996
c45202a	58	4.6	12.6087
c45202b	58	4.5	12.8889
c45210a	771	18.4	41.9022
c45211a	58	4.4	13.1818
c45220a	233	7.9	29.4937
c45220b	365	10.5	34.7619
c45220c	585	14.9	39.2617
c45220d	899	26.7	33.6704
c45220e	58	4.5	12.8889
c45220f	58	4.5	12.8889
c45641a	248	8.2	30.2439
c45641b	248	8.3	29.8795
c45641c	248	8.2	30.2439
c45662a	328	9.8	33.4694
c45662b	665	16.3	40.7975
c46011a	205	7.4	27.7027
c49020a	86	5	17.2
c49021a	60	4.6	13.0435
c52001b	152	6.2	24.5161
c53004b	85	5.1	16.6667
c53005a	58	4.5	12.8889
c53005b	367	10.5	34.9524
c53006a	58	4.6	12.6087
c53006b	511	13.3	38.421
c53007a	238	8	29.75
c54a03a	129	5.9	21.8644
c54a23a	84	4.9	17.1428
c54a26a	169	6.7	25.2239
c54a27a	82	5	16.4

c54a41a	418	11.5	36.3478
c54a42a	341	10.1	33.7624
c54a42b	330	9.8	33.6735
c54a42c	272	8.8	30.9091
c54a42e	220	7.7	28.5714
c55b03a	313	9.5	32.9474
c55b04a	164	6.5	25.2308
c55b06a	2931	84	34.8928
c55b06b	314	9.5	33.0526
c55c01a	116	5.5	21.0909
c55c02a	67	4.7	14.2553
c55c03a	58	4.4	13.1818
c55c03b	491	13	37.7692
c56002a	530	13.7	38.6861
c57002a	187	7.1	26.338
c57004a	96	5.2	18.4615
c57004b	174	6.7	25.9701
c57004c	128	5.9	21.6949
c57005a	128	5.8	22.069
c64202a	266	8.7	30.5747
c67003c	255	8.4	30.3571
c74211b	236	8	29.5
c83b02a	58	4.5	12.8889
c83b02b	80	4.9	16.3265
c83e03a	187	7.9	23.6709
c85019a	58	4.5	12.8889
c87b04c	58	4.6	12.6087
c87b15a	58	4.6	12.6087
c87b50a	66	4.7	14.0426
ca1007a1m	75	4.9	15.3061
ca1022a6m	96	5.2	18.4615
ca1105a1m	75	4.9	15.3061
ca3002a2m	83	5	16.6
ca3004e4m	136	6	22.6667
ca3006c5m	83	4.9	16.9388
ca3006d3m	75	4.8	15.625
cb1002a	58	4.4	13.1818
cc1010a	76	4.8	15.8333
fail_4660	28	3.9	7.17949
flt	66	4.7	14.0426
jfb	3222	68.6	46.9679
jrs0016	109	5.4	20.1852
jrs0021	193	7.1	27.1831
jrs0022	217	7.7	28.1818
jrs0031	264	8.6	30.6977
jrs0032	132	5.8	22.7586
jrs0050	768	18.6	41.2903
jrs0054	66	4.7	14.0426
jrs0055	48	4.3	11.1628
jrs0060	115	5.6	20.5357
jrs0070	262	8.6	30.4651
jrs0075	106	5.3	20
jrs0076	131	5.9	22.2034
jrs0077	97	5.3	18.3019
jrs0078	82	4.9	16.7347
jrs0079	74	4.7	15.7447
jrs0080	403	11.3	35.6637
jrs0096	96	5.3	18.1132
jrs0210	51	4.3	11.8605
jrs0211	54	4.5	12

jrs0240	291	9.5	30.6316
jrs0362	132	5.9	22.3729
jrs0710	79	4.9	16.1224
jrs0803	66	4.5	14.6667
jrs0810	101	5.3	19.0566
jrs0900	87	5	17.4
jrs0901	95	5.1	18.6274
jrs0902	91	5.1	17.8431
jrs0903	404	11.3	35.7522
jrs0905	196	7.1	27.6056
jrs0906	80	4.9	16.3265
jrsi906	75	4.7	15.9574
qlt	95	5.2	18.2692
rt_labels	50000	637.7	78.4068
rt_not	63	4.5	14
simple	28	3.8	7.36842

Number of tests: 190

Appendix C

Sun 4 ACVC Test Results

*** UNIX RTL ***

Assembly: SRTL/BRTL/CRTL

Program Name	Time (in seconds)
a22006b.rtl	192.8
a22006c.rtl	192.7
a22006d.rtl	192.1
a22006e.rtl	192.8
a22006f.rtl	191.9
a29002a.rtl	193.2
a29002b.rtl	192.2
a29002c.rtl	192.0
a29002d.rtl	191.8
a29002e.rtl	191.8
a29002f.rtl	191.9
a29002g.rtl	192.2
a29002h.rtl	191.6
a29002i.rtl	192.1
a29002j.rtl	191.9
a29003a.rtl	464.2
aa5008x.rtl	191.9
be_0001.rtl	78.3
be_0002.rtl	78.4
be_0003.rtl	118.9
be_0004.rtl	78.7
be_0005.rtl	78.9
be_0006.rtl	78.4
be_0007.rtl	78.5
be_0008.rtl	328.4
be_0009.rtl	254.6
be_000a.rtl	79.2
be_000b.rtl	304.4
be_000g.rtl	194.2
be_000i.rtl	175.8
be_000m.rtl	128.9
be_000y.rtl	78.8
c23001a.rtl	191.6
c23006a.rtl	192.1
c23006b.rtl	281.9
c23006c.rtl	309.4
c24002a.rtl	192.2
c24002b.rtl	192.2
c24003a.rtl	192.1
c24003b.rtl	191.8
c24113a.rtl	512.3
c24113b.rtl	512.4
c24113c.rtl	513.3
c24202a.rtl	191.7
c24202b.rtl	191.6
c24203a.rtl	191.6
c24207a.rtl	191.7
c25001a.rtl	191.8
c25001b.rtl	191.6
c25004a.rtl	201.7
c27001a.rtl	191.7
c27002a.rtl	191.7
c2a001a.rtl	192.1

c2a001b.rtl	191.6
c2a002a.rtl	192.0
c2a006a.rtl	191.3
c2a008a.rtl	191.8
c2a009a.rtl	191.2
c32203a.rtl	191.3
c34015b.rtl	284.8
c35106a.rtl	191.7
c35502g.rtl	421.1
c35503o.rtl	458.2
c35507o.rtl	327.0
c35508o.rtl	445.7
c35705a.rtl	419.2
c35705b.rtl	420.0
c35705c.rtl	420.5
c36203a.rtl	281.1
c45101a.rtl	785.0
c45101b.rtl	555.8
c45101c.rtl	409.4
c45101e.rtl	191.5
c45101g.rtl	2213.0
c45101h.rtl	1227.8
c45101i.rtl	594.6
c45101k.rtl	1333.0
c45104a.rtl	191.3
c45122b.rtl	699.2
c45122c.rtl	806.6
c45123a.rtl	348.4
c45123b.rtl	393.0
c45123c.rtl	521.2
c45201a.rtl	809.6
c45201b.rtl	874.7
c45202a.rtl	192.0
c45202b.rtl	191.6
c45210a.rtl	570.8
c45211a.rtl	191.9
c45220a.rtl	937.5
c45220b.rtl	1478.1
c45220c.rtl	1567.9
c45220d.rtl	2501.2
c45220e.rtl	191.4
c45220f.rtl	192.5
c45641a.rtl	614.3
c45641b.rtl	616.2
c45641c.rtl	614.3
c45662a.rtl	692.7
c45662b.rtl	1087.6
c46011a.rtl	597.6
c49020a.rtl	285.2
c49021a.rtl	196.4
c52001b.rtl	384.4
c53004b.rtl	277.4
c53005a.rtl	192.2
c53005b.rtl	1015.3
c53006a.rtl	192.1
c53006b.rtl	1514.8
c53007a.rtl	804.1
c53008a.rtl	742.7
c54a03a.rtl	424.5
c54a23a.rtl	284.6

c54a26a.rtl	526.5
c54a27a.rtl	266.5
c54a41a.rtl	535.8
c54a42a.rtl	1338.1
c54a42b.rtl	1839.6
c54a42c.rtl	1128.1
c54a42e.rtl	1122.9
c54a42f.rtl	1183.7
c55b03a.rtl	415.1
c55b04a.rtl	333.8
c55b06a.rtl	2521.5
c55b06b.rtl	847.5
c55c01a.rtl	347.3
c55c02a.rtl	228.5
c55c03a.rtl	191.8
c55c03b.rtl	1320.0
c56002a.rtl	1245.3
c57002a.rtl	541.4
c57004a.rtl	301.4
c57004b.rtl	568.2
c57004c.rtl	379.8
c57005a.rtl	380.2
c64202a.rtl	539.6
c67003c.rtl	669.6
c74211b.rtl	501.6
c74402b.rtl	365.1
c83b02a.rtl	191.4
c83b02b.rtl	257.3
c83e03a.rtl	432.6
c83e04a.rtl	558.1
c85019a.rtl	191.6
c87b04c.rtl	191.5
c87b15a.rtl	191.3
c87b50a.rtl	211.9
ca1003a.rtl	369.1
ca1007a1m.rtl	248.0
ca1022a6m.rtl	298.8
ca1105a1m.rtl	250.8
ca3002a2m.rtl	268.5
ca3004e4m.rtl	480.7
ca3004f4m.rtl	525.9
ca3006c5m.rtl	262.7
ca3006d3m.rtl	244.8
cb1002a.rtl	191.4
cc1010a.rtl	248.5
fail_4660.rtl	78.5
flt.rtl	199.3
jfb.rtl	355.7
jrs0016.rtl	287.8
jrs0021.rtl	821.1
jrs0022.rtl	563.4
jrs0031.rtl	411.7
jrs0032.rtl	431.0
jrs0050.rtl	476.3
jrs0054.rtl	165.9
jrs0055.rtl	127.4
jrs0060.rtl	537.3
jrs0070.rtl	615.7
jrs0075.rtl	373.2
jrs0076.rtl	571.1

jrs0077.rtl	367.3	
jrs0078.rtl	252.7	
jrs0079.rtl	249.7	
jrs0080.rtl	644.4	
jrs0096.rtl	304.6	
jrs0210.rtl	137.8	
jrs0211.rtl	144.6	
jrs0240.rtl	944.9	
jrs0362.rtl	572.0	
jrs0710.rtl	227.7	
jrs0800.rtl	178.5	
jrs0802.rtl	224.7	
jrs0803.rtl	191.6	
jrs0810.rtl	399.0	
jrs0900.rtl	282.8	
jrs0901.rtl	257.7*** ERROR ***	Error also occurs on TI VAX system, so this is a correct reults.
jrs0902.rtl	279.4	
jrs0903.rtl	370.3	
jrs0905.rtl	305.2	
jrs0906.rtl	234.8	
jrsi906.rtl	227.8	
matrix_mult_driver.rtl	279.0	
qlt.rtl	255.3	
rt_functions.rtl	171.3	
rt_jopt_test.rtl	80.1	
rt_labels.rtl	197.4	
rt_not.rtl	175.8	
rt_procedures.rtl	473.1	
sim_test_01.rtl	192.4	
sim_test_03.rtl	82.9	
sim_test_06.rtl	81.8	
simple.rtl	81.4	
sobel_driver.rtl	327.3	
vector_add.rtl	276.1	

Number of tests: 207

*** UNIX RTL ***

Linking: CRTL

Program Name	Time (in seconds)
a22006b	150.2
a22006c	150.7
a22006d	152.9
a22006e	151.1
a22006f	151.0
a29002a	152.7
a29002b	151.2
a29002c	149.4
a29002d	150.0
a29002e	150.7
a29002f	149.4
a29002g	150.1
a29002h	149.8
a29002i	150.5
a29002j	149.7
a29003a	166.0
aa5008x	149.5
be_0001	146.4
be_0002	146.3
be_0003	148.0
be_0004	146.4
be_0005	145.9
be_0006	145.8
be_0007	147.1
be_0008	153.9
be_0009	151.8
be_000a	146.3
be_000b	153.0
be_000g	149.2
be_000i	149.4
be_000m	148.1
be_000y	146.0
c23001a	150.3
c23006a	149.4
c23006b	152.8
c23006c	155.9
c24002a	149.8
c24002b	150.4
c24003a	150.2
c24003b	149.9
c24113a	159.5
c24113b	160.6
c24113c	159.9
c24202a	150.0
c24202b	149.4
c24203a	149.1
c24207a	149.9
c25001a	149.8
c25001b	149.3
c25004a	151.1
c27001a	150.0
c27002a	149.5
c2a001a	149.3

c2a001b	149.8
c2a002a	149.5
c2a006a	150.1
c2a008a	148.8
c2a009a	149.9
c32203a	149.4
c34015b	154.1
c35106a	149.4
c35502g	157.4
c35503o	160.9
c35507o	156.2
c35508o	159.2
c35705a	156.3
c35705b	155.8
c35705c	155.2
c36203a	153.7
c45101a	168.4
c45101b	161.7
c45101c	157.5
c45101e	149.5
c45101g	211.5
c45101h	179.2
c45101i	160.9
c45101k	187.2
c45104a	149.3
c45122b	168.2
c45122c	170.2
c45123a	155.6
c45123b	161.0
c45123c	160.9
c45201a	167.9
c45201b	170.9
c45202a	150.2
c45202b	149.4
c45210a	161.3
c45211a	149.4
c45220a	170.5
c45220b	187.2
c45220c	187.6
c45220d	269.7
c45220e	151.4
c45220f	151.9
c45641a	162.2
c45641b	162.0
c45641c	162.9
c45662a	163.5
c45662b	174.0
c46011a	165.1
c49020a	152.5
c49021a	149.8
c52001b	156.3
c53004b	153.1
c53005a	149.2
c53005b	177.2
c53006a	150.3
c53006b	195.5
c53007a	169.5
c53008a	169.4
c54a03a	159.5
c54a23a	160.3

c54a26a	158.0
c54a27a	152.8
c54a41a	160.9
c54a42a	187.8
c54a42b	206.5
c54a42c	178.4
c54a42e	180.3
c54a42f	182.0
c55b03a	156.3
c55b04a	154.4
c55b06a	282.9
c55b06b	171.0
c55c01a	154.9
c55c02a	150.6
c55c03a	149.2
c55c03b	183.9
c56002a	183.7
c57002a	161.2
c57004a	154.8
c57004b	163.7
c57004c	156.3
c57005a	156.0
c64202a	160.2
c67003c	168.4
c74211b	162.5
c74402b	156.8
c83b02a	149.0
c83b02b	151.2
c83e03a	156.6
c83e04a	161.0
c85019a	149.1
c87b04c	148.5
c87b15a	149.0
c87b50a	150.8
ca1003a	156.0
ca1007a1m	151.3
ca1022a6m	153.8
ca1105a1m	152.3
ca3002a2m	161.2
ca3004e4m	157.7
ca3004f4m	159.7
ca3006c5m	152.9
ca3006d3m	157.0
cb1002a	148.9
cc1010a	151.0
fail_4660	145.2
flt	148.9
jfb	155.2
jrs0016	152.5
jrs0021	167.7
jrs0022	159.6
jrs0031	155.8
jrs0032	155.5
jrs0050	156.3
jrs0054	148.5
jrs0055	147.7
jrs0060	159.9
jrs0070	160.4
jrs0075	153.3
jrs0076	158.9

```
jrs0077                                153.7
jrs0078                                150.1
jrs0079                                151.0
jrs0080                                160.8
jrs0096                                152.3
jrs0210                                146.8
jrs0211                                146.0
jrs0240                                167.7
jrs0362                                159.8
jrs0710                                150.1
jrs0800                                148.2
jrs0802                                149.3
jrs0803                                148.5
jrs0810                                154.7
jrs0900                                151.7
jrs0901                                150.2
jrs0902                                151.1
jrs0903                                155.4
jrs0905                                154.2
jrs0906                                150.3
jrsi906                                151.6
matrix_mult_driver                       0.0
qlt                                    153.6
rt_functions                           151.2
rt_jopt_test                           148.1
rt_labels                              151.3
rt_not                                 152.6
rt_procedures                          162.1
sim_test_01                            163.6
sim_test_03                            156.1
sim_test_06                            151.8
simple                                 151.3
sobel_driver                           161.2
vector_add                             159.9
-----------------------------------------------
Number of tests: 207
```

*** UNIX RTL ***

Simulation

Program Name	Cycles	Time (in sec)	Cycs/Sec
a22006b	58	3.5	16.6
a22006c	58	3.5	16.6
a22006d	58	3.4	17.1
a22006e	58	3.4	17.1
a22006f	58	3.5	16.6
a29002a	58	3.6	16.1
a29002b	58	3.5	16.6
a29002c	58	3.6	16.1
a29002d	58	3.4	17.1
a29002e	58	3.4	17.1
a29002f	58	3.5	16.6
a29002g	58	3.5	16.6
a29002h	58	3.5	16.6
a29002i	58	3.5	16.6
a29002j	58	3.5	16.6
a29003a	226	6.6	34.2
aa5008x	58	3.5	16.6
be_0001	28	2.9	9.7
be_0002	28	2.9	9.7
be_0003	46	3.3	13.9
be_0004	28	3.0	9.3
be_0005	28	2.9	9.7
be_0006	28	3.0	9.3
be_0007	28	2.9	9.7
be_0008	323	8.2	39.4
be_0009	101	4.5	22.4
be_000a	28	2.9	9.7
be_000b	118	4.6	25.7
be_000g	646	14.1	45.8
be_000i	315	8.2	38.4
be_000m	51	3.4	15.0
be_000y	28	3.0	9.3
c23001a	58	3.5	16.6
c23006a	58	3.4	17.1
c23006b	91	4.0	22.8
c23006c	101	4.3	23.5
c24002a	58	3.6	16.1
c24002b	58	3.4	17.1
c24003a	58	3.5	16.6
c24003b	58	3.5	16.6
c24113a	182	6.0	30.3
c24113b	182	5.9	30.8
c24113c	182	6.1	29.8
c24202a	58	3.5	16.6
c24202b	58	3.5	16.6
c24203a	58	3.5	16.6
c24207a	58	3.5	16.6
c25001a	58	3.5	16.6
c25001b	58	3.4	17.1
c25004a	62	3.7	16.8
c27001a	58	3.5	16.6
c27002a	58	3.6	16.1
c2a001a	58	3.5	16.6

c2a001b	58	3.4	17.1
c2a002a	58	3.6	16.1
c2a006a	58	3.5	16.6
c2a008a	58	3.5	16.6
c2a009a	58	3.5	16.6
c32203a	58	3.5	16.6
c34015b	89	4.1	21.7
c35106a	58	3.5	16.6
c35502g	381	9.3	41.0
c35503o	158	5.3	29.8
c35507o	106	4.5	23.6
c35508o	153	5.3	28.9
c35705a	152	5.6	27.1
c35705b	152	5.6	27.1
c35705c	152	5.5	27.6
c36203a	88	4.0	22.0
c45101a	1404	27.7	50.7
c45101b	710	15.4	46.1
c45101c	271	7.4	36.6
c45101e	58	3.4	17.1
c45101g	4938	92.5	53.4
c45101h	2222	42.5	52.3
c45101i	519	11.9	43.6
c45101k	2275	44.0	51.7
c45104a	58	3.4	17.1
c45122b	1194	25.6	46.6
c45122c	440	10.6	41.5
c45123a	264	7.2	36.7
c45123b	271	7.5	36.1
c45123c	175	5.7	30.7
c45201a	11616	214.4	54.2
c45201b	9774	181.3	53.9
c45202a	58	3.5	16.6
c45202b	58	3.4	17.1
c45210a	771	17.1	45.1
c45211a	58	3.5	16.6
c45220a	233	6.7	34.8
c45220b	365	9.3	39.2
c45220c	585	13.1	44.7
c45220d	899	24.5	36.7
c45220e	58	3.6	16.1
c45220f	58	3.5	16.6
c45641a	248	7.2	34.4
c45641b	248	7.2	34.4
c45641c	248	7.1	34.9
c45662a	328	11.0	29.8
c45662b	665	14.7	45.2
c46011a	205	6.1	33.6
c49020a	86	4.0	21.5
c49021a	60	3.5	17.1
c52001b	152	5.4	28.1
c53004b	85	4.0	21.3
c53005a	58	3.6	16.1
c53005b	367	9.1	40.3
c53006a	58	3.5	16.6
c53006b	511	11.8	43.3
c53007a	238	7.1	33.5
c54a03a	129	5.0	25.8
c54a23a	84	4.3	19.5
c54a26a	169	5.6	30.2

c54a27a	82	4.0	20.5
c54a41a	418	10.6	39.4
c54a42a	341	8.7	39.2
c54a42b	330	8.5	38.8
c54a42c	272	7.4	36.8
c54a42e	220	6.6	33.3
c55b03a	313	8.2	38.2
c55b04a	164	5.5	29.8
c55b06a	2931	76.4	38.4
c55b06b	314	8.3	37.8
c55c01a	116	4.6	25.2
c55c02a	67	3.7	18.1
c55c03a	58	3.5	16.6
c55c03b	491	11.5	42.7
c56002a	530	11.8	44.9
c57002a	187	5.8	32.2
c57004a	96	5.4	17.8
c57004b	174	5.6	31.1
c57004c	128	5.0	25.6
c57005a	128	5.0	25.6
c64202a	266	7.3	36.4
c67003c	255	7.0	36.4
c74211b	236	6.7	35.2
c83b02a	58	3.5	16.6
c83b02b	80	4.1	19.5
c83e03a	187	6.2	30.2
c85019a	58	3.5	16.6
c87b04c	58	3.5	16.6
c87b15a	58	3.4	17.1
c87b50a	66	3.7	17.8
ca1007a1m	75	4.0	18.8
ca1022a6m	96	5.2	18.5
ca1105a1m	75	3.9	19.2
ca3002a2m	83	4.1	20.2
ca3004e4m	136	5.1	26.7
ca3006c5m	83	4.0	20.8
ca3006d3m	75	3.9	19.2
cb1002a	58	3.5	16.6
cc1010a	76	4.0	19.0
fail_4660	28	2.9	9.7
flt	66	3.6	18.3
jfb	3222	62.6	51.5
jrs0016	109	4.5	24.2
jrs0021	193	6.0	32.2
jrs0022	217	6.4	33.9
jrs0031	264	7.4	35.7
jrs0032	132	4.9	26.9
jrs0050	768	17.4	44.1
jrs0054	66	3.7	17.8
jrs0055	48	3.4	14.1
jrs0060	115	4.8	24.0
jrs0070	262	7.3	35.9
jrs0075	106	4.3	24.7
jrs0076	131	4.9	26.7
jrs0077	97	4.4	22.0
jrs0078	82	4.0	20.5
jrs0079	74	4.6	16.1
jrs0080	403	9.8	41.1
jrs0096	96	4.2	22.9
jrs0210	51	3.4	15.0

jrs0211	54	3.5	15.4
jrs0240	291	8.0	36.4
jrs0362	132	4.8	27.5
jrs0710	79	3.9	20.3
jrs0803	66	3.7	17.8
jrs0810	101	4.3	23.5
jrs0900	87	4.1	21.2
jrs0901	95	4.2	22.6
jrs0902	91	4.1	22.2
jrs0903	404	9.9	40.8
jrs0905	196	7.0	28.0
jrs0906	80	3.9	20.5
jrsi906	75	3.9	19.2
matrix_mult_driver	21768	746.3	28.5
qlt	95	5.0	19.0
rt_functions	4	61.0	0.1
rt_jopt_test	28	3.0	9.3
rt_labels	4	73.0	0.1
rt_not	3	3.5	0.9
rt_procedures	4	90.0	0.0
sim_test_01	4	69.0	0.1
sim_test_03	4	28.0	0.1
sim_test_06	4	28.0	0.1
simple	8	3.0	2.7
sobel_driver	1783	38.1	46.8

Number of tests: 198

Appendix D

Characterization of VHDL Features

Performance Characteristics of the Intermetrics 1076 Support Environment

1 Introduction

This study was undertaken at the request of JRS. To support their effort to rehost the Intermetrics 1076 Support Environment to the IBM PC, JRS wished to know more about performance characteristics of the toolset. This knowledge would aid JRS in characterizing the types of VHDL models that a user could expect to run, with reasonable performance, on the toolset when it had been rehosted to the IBM PC.

The purpose of the study was to determine how the use of particular VHDL language features of VHDL affects performance of the Intermetrics 1076 Support Environment hosted on the Sun3. Determining the capacity limits of the toolset was not within the scope of the study.

2 Method

From previous internal Intermetrics studies, it was known that the concurrent features of VHDL have a more interesting effect on performance than do the sequential features of VHDL.

We looked at the following features:

a. signal declarations

b. component instantiations and port maps

c. blocks statements

d. generate statements

e. process statements

f. signal assignment statements

For these features, we examined the following characteristics:

a. speed of analysis

b. speed of model generation

c. speed of the build program

d. speed of simulation

e. size of IVAN

f. size of compiled C object

g. size of kernel

h. size of signal dictionary

i. average memory usage during analysis

j. average memory usage during model generation

k. average memory usage during building

l. average memory usage during simulation

m. source lines of code

The general method was to devise test cases that isolated as much as possible the feature under investigation. The feature under investigation was replicated in the test case some number of times, either by textual replication in the VHDL source or by replication via elaboration of a generate statement.

All tests were run on a Sun 3/60 with 8 MB memory.

Measurements were taken as follows:

a. Speed measurements were taken from the %U option on the csh variable "time"; this option gives amount of cpu seconds devoted to a user process.

b. Size of disk-resident data (IVAN, compiled C object, kernel, signal dictionary) was taken from the "ls -l" command.

c. Internal memory usage was taken from the %D option on the csh variable "time"; this option gives average amount of unshared data space used by a program.

d. Source lines of code (SLOC) was measured by counting all semicolons and commas in the VHDL text.

The following table lists the feature tested by each of the thirty-one tests that were retained in the final version of the set of tests.

VHDL Performance Characteristics
Intermetrics, Inc.

Feature	Test	N	
unres. integer signal	t_0	100	
res. integer signal	t_00	100	
empty component	t_1	100	
port associations			
mode in, unres.	t_2	100	(10 comp. inst., 1 sig. decl.)
mode out, unres.	t_3	100	(10 comp. inst., 100 sig. decl.)
mode inout, unres.	t_4	100	(10 comp. inst., 100 sig. decl.)
mode in, res.	t_5	100	(10 comp. inst., 1 res. sig. decl.)
mode out, res.	t_6	100	(10 comp. inst., 100 res. sig. decl.)
mode inout, res.	t_7	100	(10 comp. inst., 100 res. sig. decl.)
mode out, res.	t_8	100	(10 comp. inst., 1 res. sig. decl.)
mode inout, res.	t_9	100	(10 comp. inst., 1 res. sig. decl.)
processes			
empty process with one wait	t_10	100	
seq. sig. assig. and wait	t_11	100	(100 sig. decl.)
hierarchy elaboration - all generated			
generate label	t_40		(10000 generate loops)
4 levels, no blocks, no ports, no sigs	t_41		(1110 comp. inst., 1110 generate label inst.)
4 levels, no sigs, no ports	t_43		(2100 block inst., 1110 comp. inst. 1110 generate label inst.)
4 levels, sigs, no ports	t_44		(like t_43 plus 5000 unres. signals)
4 levels, sigs, conc. assign, no ports	t_45		(like t_44 plus 5000 concurrent signal assignments)
4 levels, sigs, conc. assign, ports	t_46		(like t_45 plus 5550 ports mode in)
4 levels, no blocks, no ports, sigs	t_47		(like t_41 plus 5000 signals)
2 levels, blocks, sigs, no ports	t_48		(1000 generated blocks, 1000 instances, 10000 sigs)

simulation tests - all generated			
elab. empty comp.	t_50	10000	
elab. empty block.	t_51	10000	
elab. comp. and sig.	t_59	10000	
elab. block and sig.	t_52	10000	
ports			
in, unres.	t_53	10000	(1000 comp. inst., 10000 sig. decl.)
out, unres.	t_54	10000	(1000 comp. inst., 10000 sig. decl.)
inout, unres.	t_55	10000	(1000 comp. inst., 10000 sig. decl.)
in, res.	t_56	10000	(1000 comp. inst., 10000 sig. decl.)
out, res.	t_57	10000	(1000 comp. inst., 10000 sig. decl.)
inout, res.	t_58	10000	(1000 comp. inst., 10000 sig. decl.)

3 Results

The table of actual results for the thirty-one test cases are appended to this report. This section summarizes the results.

3.1 General Observations

It would be convenient if one were able to predict performance of the toolset from the SLOC count. Unfortunately, as the following figures show, the performance of the Analyzer and especially of the Model Generator is dependent on the kind of VHDL under consideration:

a. The size of the IVAN varied from 0.25 KB per SLOC to 0.50 KB per SLOC.

b. Analyzer speed varied from 15 SLOC per cpu second to 30 SLOC per cpu second for files containing more than 100 SLOC (for flies with fewer than 100 SLOC, figures are unreliable due to the amount of startup time required by the Analyzer).

c. Model Generator speed varied from 0.5 SLOC per cpu second to 1.7 SLOC per cpu second.

d. The size of the compiled object varied from 0.25 KB per SLOC to 1.3 KB per SLOC.

Similar considerations apply to the other measurements. Performance predictions from SLOC counts for heterogeneous VHDL samples are not reliable.

3.2 Details

The following figures for individual VHDL features were calculated from the test cases by first subtracting the time (or size) not associated with the feature in question from the total time (or size) for the test case and then dividing the result by the number of replications of the feature. For example, to compute the simulation time required for one port association of mode "in", the time for test case t_59 was subtracted from the time for test case t_48 (which differed from t_53 in having instantiations of subcomponents with no ports) and the result divided by 10,000.

	ivan kb	obj kb	alz cpu	mg cpu	alz mem	mg mem
1 unres. int. sig. decl.	0.21	0.28	0.04	0.72	1.28	1.84
1 res. int. sig. decl.	0.21	0.35	0.04	0.80	1.68	1.84
1 empty comp. inst.	0.25	0.48	0.04	0.79	1.44	1.68
port assocs						
1 mode in, unres.	0.05	0.05	0.03	0.14	1.44	0.69
1 mode out, unres.	0.06	0.27	0.03	0.90	0.50	1.35
1 mode inout, unres.	0.06	0.30	0.03	1.00	0.58	1.99
1 mode in, res.	0.05	0.05	0.03	0.15	1.44	0.77
1 mode out, res.	0.06	0.24	0.03	0.45	0.18	0.47
1 mode inout, res.	0.06	0.27	0.03	0.54	0.18	0.55
1 mode out, res.	0.05	0.24	0.03	0.44	1.52	0.85
1 mode inout, res.	0.05	0.27	0.03	0.53	1.44	0.93
processes						
empty, one wait	0.29	0.44	0.03	0.79	1.28	0.96
seq. sig. assign, 1 wait	0.32	0.63	0.07	2.16	0.88	2.24

	sim cpu	sd kb	sim mem
hierarchy			
1 block label	0.004	0.023	0.030
1 sig decl	0.001	0.036	0.038
1 conc. sig. assign	0.002	0.044	0.032
1 unres. in port	0.001	0.036	0.007
simulation			
generate loop	0.022	0.053	0.061
empty component	0.007	0.097	0.075
empty block	0.022	0.053	0.057
signal decl.	0.001	0.036	0.047
unresolved in port	0.003	0.036	0.022
unresolved out port	0.001	0.036	0.009
unresolved inout port	0.001	0.036	0.003
resolved in port	0.003	0.036	0.022
resolved out port	0.002	0.036	0.071
resolved inout port	0.002	0.036	0.074

3.3 Complicating Factors

Two factors tend to obscure the relationship between particular VHDL features and performance of the toolset.

One factor is the size of identifiers. Both the size of the IVAN and the size of the signal dictionary depend on the length of identifiers used in the VHDL source text.

The other factor is an anomaly in the simulation performance of empty generate statements. Test t_40, which consists of the elaboration of an empty generate statement that loops 10,000 times, t_40 required 222.5 cpu seconds to simulate. Test t_50 is identical to test t_40 except that in t_50 the generate statement encloses a component instantiation; however, t_50 requires only 69.2 seconds to simulate. It is contrary to expectations that the more complex operation should require less cpu time. Profiling the run of t_40 reveals that most of the time is spent in the library routine "free," which frees space that was temporarily allocated during each of the 10,000 loops; however, we do not know why the particular memory allocation/deallocation pattern used in t_40 results in such inefficient performance. In any case, this anomaly points to unexplained behavior which may be contaminating the test results.

Unit	ivan kb	obj kb	kern kb	alz cpu	mg cpu	bld cpu	sim cpu	sd kb	sloc	alz mem	mg mem	bld mem	sim mem
empty arch.	4	2	238	2.1	5.6	8.6	1.1	8	1	792	288	264	216
t_0.vhd	25	30	254	5.7	77.2	9.1	1.0	12	101	920	472	280	232
t_00.vhd	25	37	262	6.5	85.8	9.5	1.0	12	103	960	472	280	232
t_1.vhd	29	50	279	5.8	84.3	9.4	1.2	12	103	936	456	272	224
t_2.vhd	12	12	246	5.4	28.5	9.1	1.1	12	24	952	376	280	232
t_3.vhd	33	62	287	8.9	174.6	9.4	1.0	16	123	984	624	280	224
t_4.vhd	33	65	287	9.0	185.3	10.0	1.2	16	123	992	688	288	232
t_5.vhd	12	12	262	5.8	29.2	9.6	1.1	12	26	952	384	280	232
t_6.vhd	33	66	295	10.0	138.3	10.4	1.3	16	125	992	536	288	232
t_7.vhd	33	69	303	9.5	147.3	10.1	1.2	16	125	992	544	288	240
t_8.vhd	12	31	270	5.7	58.1	10.1	1.0	12	26	960	392	288	232
t_9.vhd	12	34	279	5.8	67.1	9.9	1.2	12	26	952	400	280	232
t_10.vhd	33	46	279	5.4	84.2	9.2	1.1	12	101	920	384	280	224
t_11.vhd	57	93	311	13.1	293.1	9.2	1.1	16	401	1008	696	280	216
t_40.vhd	4	2	238	1.9	7.0	8.8	222.5	528	2	832	320	272	608
t_41.vhd	4	3	254	2.7	8.0	9.8	4.4	111	5	856	328	280	264
t_43.vhd	4	3	254	2.9	7.8	9.9	12.7	160	5	856	328	280	328
t_44.vhd	4	3	254	2.8	8.0	9.9	11.3	385	5	864	328	280	456
t_45.vhd	4	3	254	2.7	8.1	10.1	19.2	606	5	856	328	280	616
t_46.vhd	8	4	254	2.9	9.6	9.4	22.4	807	7	888	344	280	656
t_47.vhd	4	3	254	2.8	7.8	9.9	9.5	291	5	856	328	280	456
t_48.vhd	8	6	238	3.2	12.8	8.7	22.1	508	16	880	344	280	776
t_50.vhd	4	3	238	3.0	7.9	9.1	69.2	971	5	864	336	272	752
t_51.vhd	4	2	238	2.2	7.1	8.8	224.9	528	3	832	320	272	568
t_52.vhd	4	3	238	2.4	7.9	8.5	699.4	1335	4	848	328	272	752
t_53.vhd	12	7	246	3.9	15.6	9.6	54.0	868	26	912	360	280	992

t_54.vhd	12	9	246	4.1	24.5	9.8	31.9	868	26	912	360	280	864
t_55.vhd	12	9	246	3.8	25.6	10.2	27.3	868	26	920	360	280	808
t_56.vhd	12	8	254	4.6	17.7	10.1	53.6	868	29	928	376	280	1000
t_57.vhd	12	9	254	4.4	19.7	9.5	41.2	868	28	936	368	280	1488
t_58.vhd	12	10	254	4.5	21.2	10.0	42.1	868	28	920	376	280	1512
t_59.vhd	4	3	238	2.8	7.9	9.5	77.6	1335	5	864	328	280	1224
1 empty architecture	4.00	2.00	238	2.10	5.60	8.6	1.10	8	1	792	288	264	216
1 unres. int. sig. decl.	0.21	0.28		0.04	0.72					1.28	1.84		
1 res. int. sig. decl.	0.21	0.35		0.04	0.80					1.68	1.84		
1 empty comp. inst.	0.25	0.48		0.04	0.79					1.44	1.68		
port assocs													
1 mode in, unres.	0.05	0.05		0.03	0.14					1.44	0.69		
1 mode out, unres.	0.06	0.27		0.03	0.90					0.50	1.35		
1 mode inout, unres.	0.06	0.30		0.03	1.00					0.58	1.99		
1 mode in, res.	0.05	0.05		0.03	0.15					1.44	0.77		
1 mode out, res.	0.06	0.24		0.03	0.45					0.18	0.47		
1 mode inout, res.	0.06	0.27		0.03	0.54					0.18	0.55		
1 mode out, res. (1 sig	0.05	0.24		0.03	0.44					1.52	0.85		
1 mode inout, res. (1 s	0.05	0.27		0.03	0.53					1.44	0.93		
processes													
empty, one wait	0.29	0.44		0.03	0.79					1.28	0.96		
seq. sig. assign, 1 wai	0.32	0.63		0.07	2.16					0.88	2.24		
hierarchy													
1 block label							0.004	0.023					0.030
1 sig decl							0.001	0.036					0.038
1 conc. sig. assign							0.002	0.044					0.032
1 unres. in port							0.001	0.036					0.007
simulation													
generate loop							0.022	0.053					0.061
empty component							0.007	0.097					0.075
empty block							0.022	0.053					0.057
signal decl.							0.001	0.036					0.047
unresolved in port							0.003	0.036					0.022
unresolved out port							0.001	0.036					0.009
unresolved inout port							0.001	0.036					0.003
resolved in port							0.003	0.036					0.022
resolved out port							0.002	0.036					0.071
resolved inout port							0.002	0.036					0.074